1. MANUAL TECNICO

1.1. Objetivos técnicos

Este manual esta dirigido a toda persona que tenga conocimiento de desarrollo en aplicaciones WEB, el acceso a la información del Modulo solo será posible para los usuarios autorizados que estén vinculadas directamente con el área de sistemas, esto es posible mediante la asignación de usuario y contraseña controlada por el modulo de sguridades.

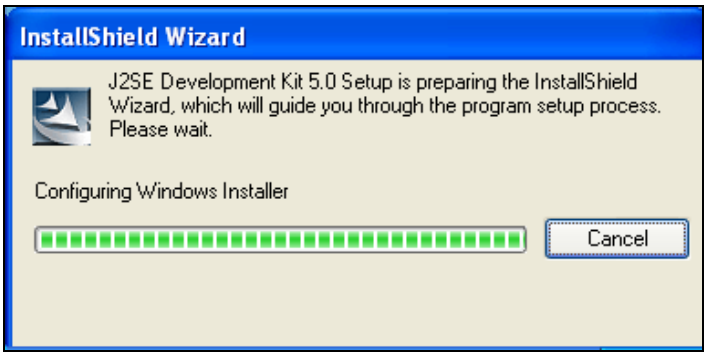
1.2. Instalación del J2SE 5.0 JDK (JAVA 2 STANDARD EDITION 5.0 JAVA DEVELOPMENT KIT).

El J2SE 5.0 JDK, sirve para crear aplicaciones JAVA.

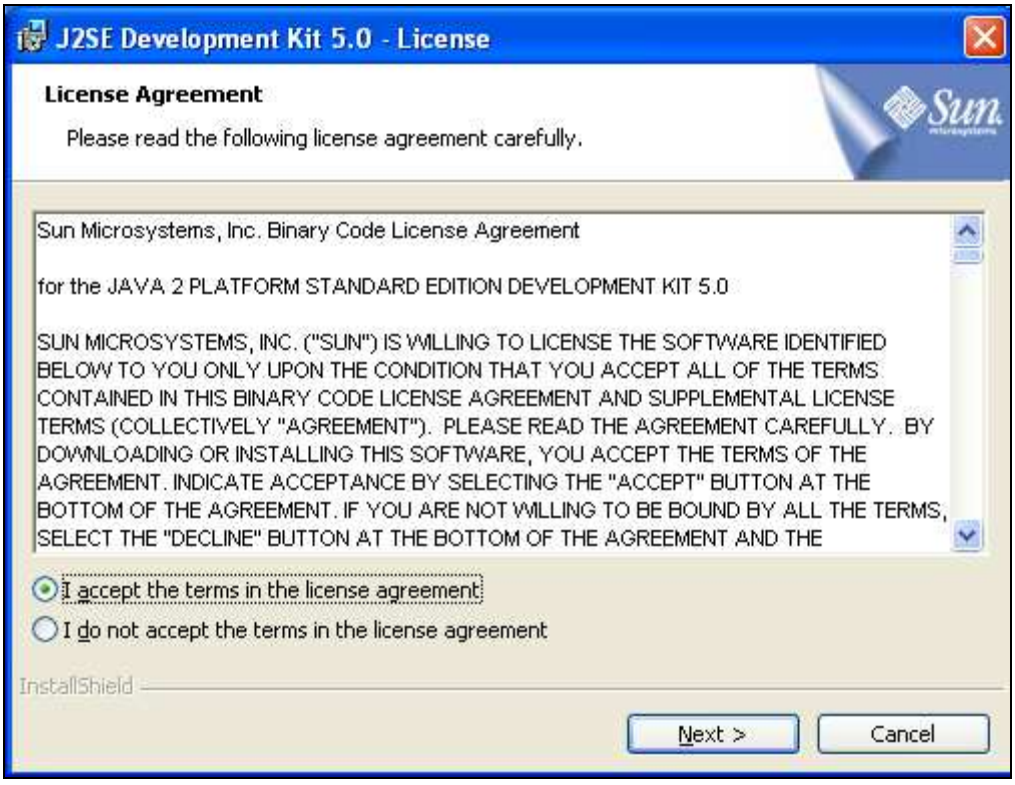
Para proceder a la instalación del J2SE 5.0 JDK también conocido como SDK (Software Development Kit), del entorno de ejecución de Java 2, Edición Estándar (JRE), se requiere poseer el instalador del J2SE 5.0 JDK (jdk-1_5_0-windows-i586.exe) para S.O. Windows el cual se lo puede descargar gratuitamente desde la propia página web de Sun Microsystems (http://java.sun.com/javase/downloads/index_jdk5.jsp).

A continuación se detallan los pasos a seguir para la instalación del J2SE 5.0 JDK en sistemas operativo Windows XP.

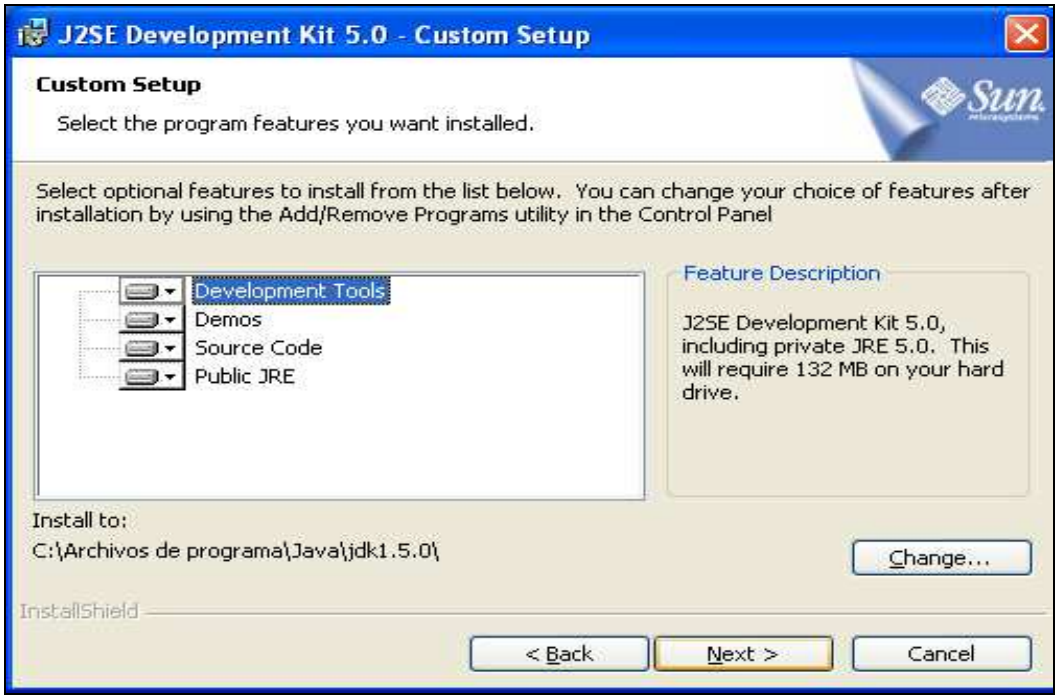
Ejecutamos el archivo que descargamos de la página web mencionada anteriormente. Se realiza la ejecución del archivo jdk-1_5_0-windows-i586.exe, una vez ejecutado se comienza a extraer los archivos de instalación y se presentan las siguientes pantallas.


InstallShield Wizard
J2SE Development Kit 5.0 Setup is preparing the InstallShield Wizard, which will guide you through the program setup process. Please wait.
Configuring Windows Installer
Cancel

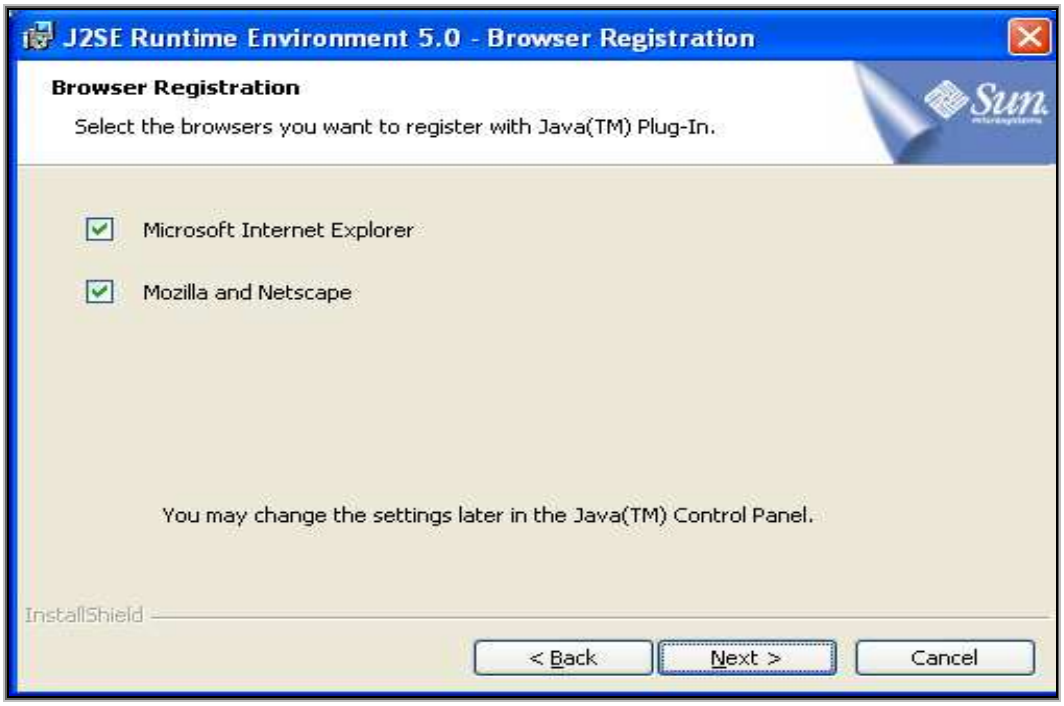
Seleccionamos la opción que indica que acepta los términos de la licencia (I accept the terms in the license agreement). Presionamos el botón Next para obtener la siguiente pantalla.


J2SE Development Kit 5.0 - License
License Agreement
Please read the following license agreement carefully.
Sun
Sun Microsystems, Inc. Binary Code License Agreement
for the JAVA 2 PLATFORM STANDARD EDITION DEVELOPMENT KIT 5.0
SUN MICROSYSTEMS, INC. ("SUN") IS WILLING TO LICENSE THE SOFTWARE IDENTIFIED BELOW TO YOU ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS BINARY CODE LICENSE AGREEMENT AND SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY "AGREEMENT"). PLEASE READ THE AGREEMENT CAREFULLY. BY DOWNLOADING OR INSTALLING THIS SOFTWARE, YOU ACCEPT THE TERMS OF THE AGREEMENT. INDICATE ACCEPTANCE BY SELECTING THE "ACCEPT" BUTTON AT THE BOTTOM OF THE AGREEMENT. IF YOU ARE NOT WILLING TO BE BOUND BY ALL THE TERMS, SELECT THE "DECLINE" BUTTON AT THE BOTTOM OF THE AGREEMENT AND THE
I accept the terms in the license agreement
I do not accept the terms in the license agreement
InstallShield
Next >
Cancel

1. Escogemos la opción de Herramientas de Desarrollo (Development Tools). Presionamos el botón Next, en la que dicha acción instalará el J2SE Development Kit 5.0 para obtener la siguiente pantalla.


J2SE Development Kit 5.0 - Custom Setup
Custom Setup
Select the program features you want installed.
Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel
Development Tools
Demos
Source Code
Public JRE
Feature Description
J2SE Development Kit 5.0, including private JRE 5.0. This will require 132 MB on your hard drive.
Install to:
C:\Archivos de programa\Java\jdk1.5.0\
Change...
InstallShield
< Back
Next >
Cancel

2. Seleccionamos el navegador de internet en el que deseamos registrar la instalación realizada. Presionamos el botón Next para obtener la siguiente pantalla.


J2SE Runtime Environment 5.0 - Browser Registration
Browser Registration
Select the browsers you want to register with Java(TM) Plug-In.
Microsoft Internet Explorer
Mozilla and Netscape
You may change the settings later in the Java(TM) Control Panel.
InstallShield
< Back
Next >
Cancel

3. Luego de la instalación de J2SE 5.0 JDK aparecerá una ventana indicando que el proceso de instalación ha finalizado. Luego presione el botón Finish.

4. Con esto finaliza la instalación del J2SE 5.0 JDK.

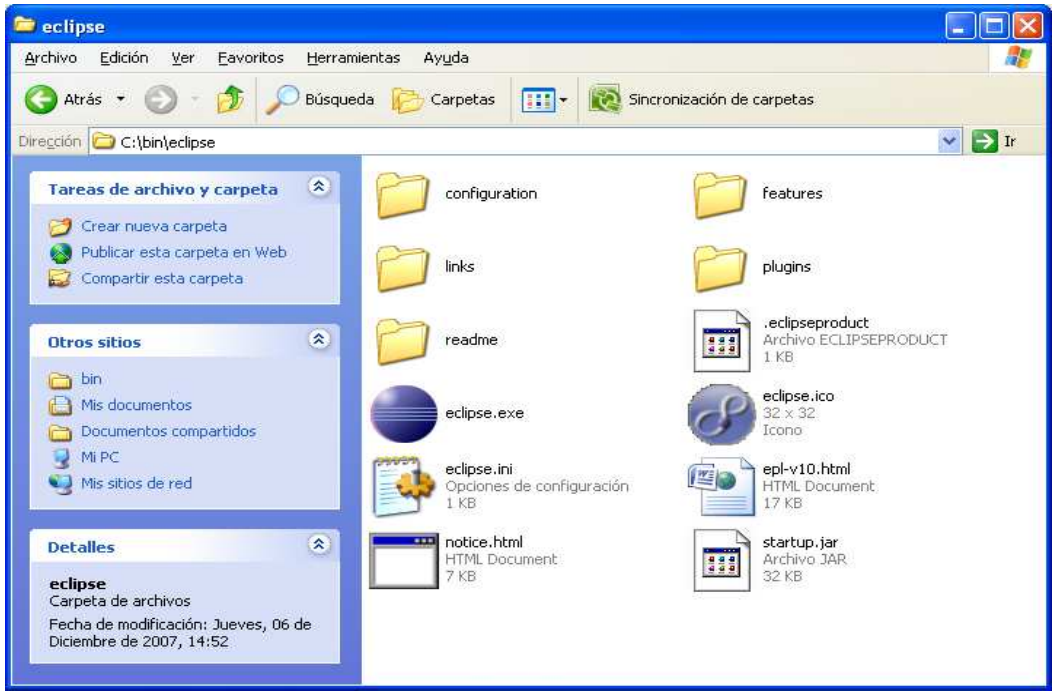
1.3. Instalación Del Entorno De Desarrollo Integrado Eclipse

Un entorno de desarrollo integrado (IDE), puede hacernos el trabaja más sencillo, sobre todo si nuestro desarrollo maneja un buen número de clases. Eclipse es un IDE de código abierto.

Para proceder a la instalación del IDE Eclipse en su versión 3.1.2 se requiere poseer el archivo ZIP (eclipse-SDK-3.1.2-win32.zip) el cual se lo puede

descargar gratuitamente desde la propia página web de eclipse (www.eclipse.org).

Su instalación es sencilla, solo tenemos que descomprimir el archivo ZIP en el directorio en donde deseamos tenerlo instalado. Crearemos una carpeta con nombre bin en la unidad C, obteniendo el directorio C:\bin, en el cual vamos descomprimir el archivo ZIP, como se muestra en la pantalla.



1.4. Instalación Del Entorno De Desarrollo Integrado My eclipse

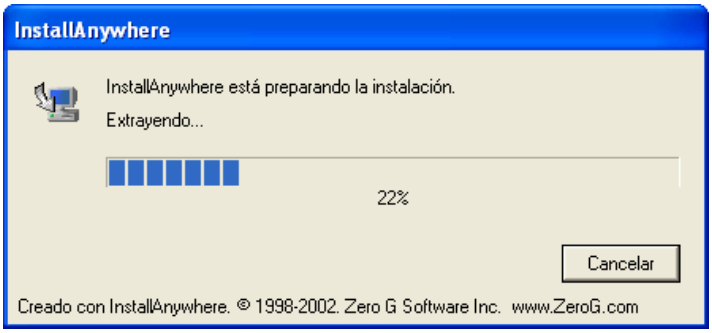
MyEclipse es un IDE que se basa en la plataforma de Eclipse.

Para proceder a la instalación del IDE MyEclipse en su versión 4.1.1, se requiere poseer el archivo ejecutable de MyEclipse o instalador

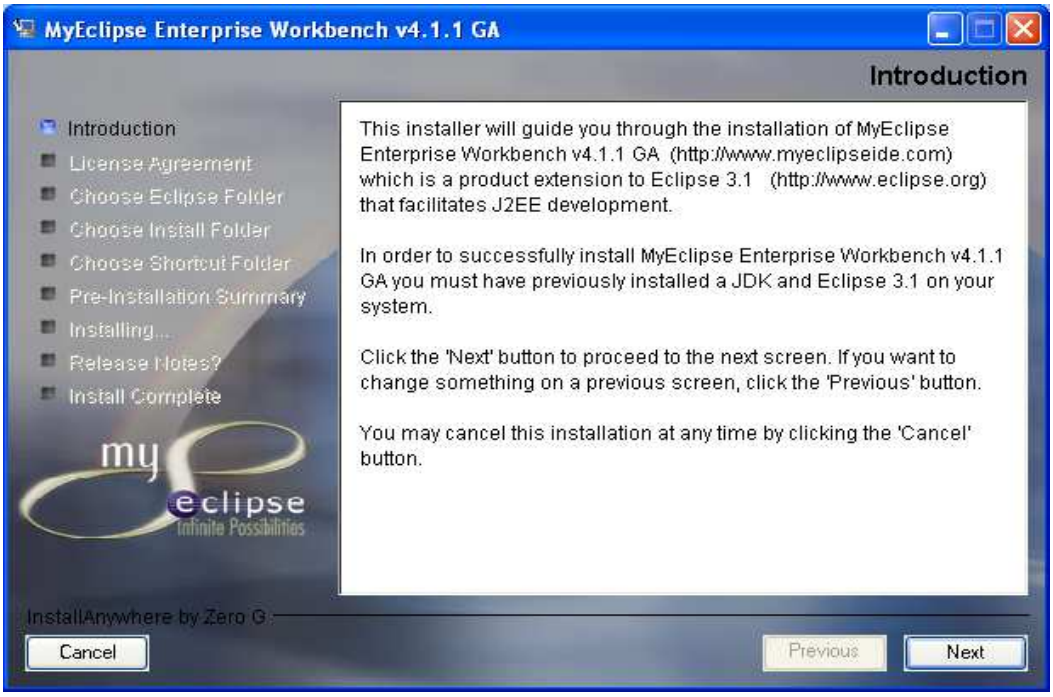
(EnterpriseWorkbenchInstaller_4.1.1GA_E3.1.exe) el cual se lo puede descargar gratuitamente desde la propia página web de MyEclipse (http://www.myeclipseide.com/).

A continuación se detallan los pasos a seguir para la instalación del IDE MyEclipse 4.1.1 en sistema operativo Windows XP.

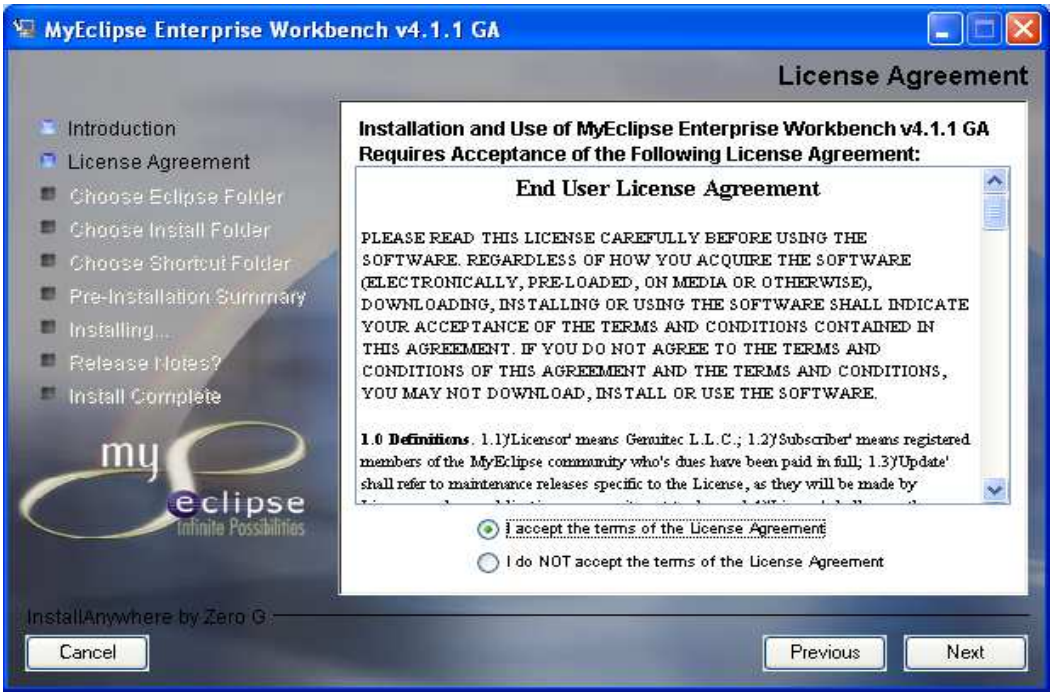
1. Ejecutamos el archivo que descargamos de la página web mencionada anteriormente. Se realiza la ejecución del archivo EnterpriseWorkbenchInstaller_4.1.1GA_E3.1.exe, una vez ejecutado se comienza a extraer los archivos de instalación y se presenta la siguiente pantalla.



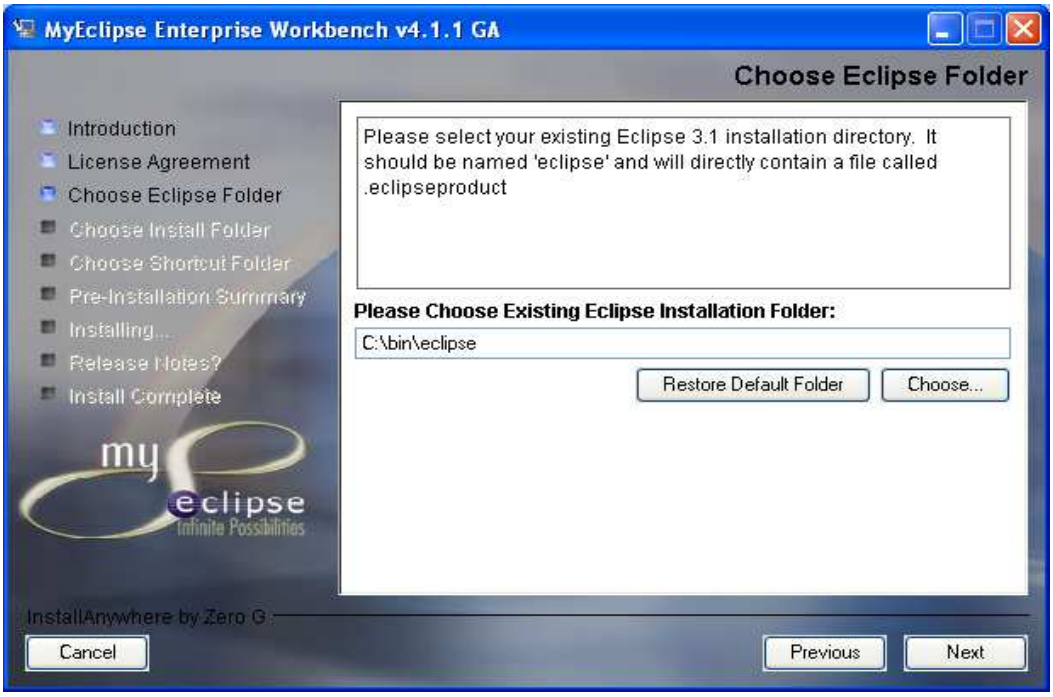
2. Bienvenidos a la instalación de MyEclipse. Presionamos el botón Next para obtener la siguiente pantalla.


MyEclipse Enterprise Workbench v4.1.1 GA
Introduction
Introduction
License Agreement
Choose Eclipse Folder
Choose Install Folder
Choose Shortcut Folder
Pre-Installation Summary
Installing...
Release Notes?
Install Complete
This installer will guide you through the installation of MyEclipse Enterprise Workbench v4.1.1 GA (http://www.myeclipseide.com) which is a product extension to Eclipse 3.1 (http://www.eclipse.org) that facilitates J2EE development.
In order to successfully install MyEclipse Enterprise Workbench v4.1.1 GA you must have previously installed a JDK and Eclipse 3.1 on your system.
Click the 'Next' button to proceed to the next screen. If you want to change something on a previous screen, click the 'Previous' button.
You may cancel this installation at any time by clicking the 'Cancel' button.
InstallAnywhere by Zero G
Cancel
Previous
Next

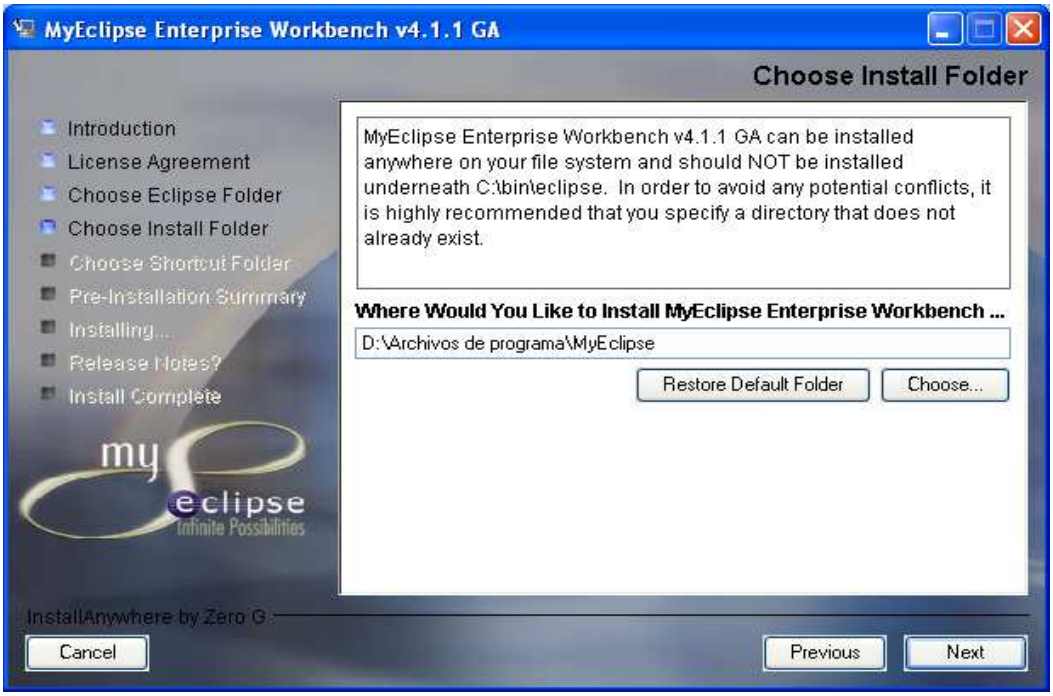
3. Seleccionamos la opción que indica que acepta los términos de la licencia (I accept the terms of the License Agreement). Presionamos el botón Next para obtener la siguiente pantalla.


MyEclipse Enterprise Workbench v4.1.1 GA
License Agreement
Introduction
License Agreement
Choose Eclipse Folder
Choose Install Folder
Choose Shortcut Folder
Pre-Installation Summary
Installing...
Release Notes?
Install Complete
Installation and Use of MyEclipse Enterprise Workbench v4.1.1 GA Requires Acceptance of the Following License Agreement:
End User License Agreement
PLEASE READ THIS LICENSE CAREFULLY BEFORE USING THE SOFTWARE. REGARDLESS OF HOW YOU ACQUIRE THE SOFTWARE (ELECTRONICALLY, PRE-LOADED, ON MEDIA OR OTHERWISE), DOWNLOADING, INSTALLING OR USING THE SOFTWARE SHALL INDICATE YOUR ACCEPTANCE OF THE TERMS AND CONDITIONS CONTAINED IN THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT AND THE TERMS AND CONDITIONS, YOU MAY NOT DOWNLOAD, INSTALL OR USE THE SOFTWARE.
1.0 Definitions. 1.1)'Licensor' means Genuitec L.L.C.; 1.2)'Subscriber' means registered members of the MyEclipse community who's dues have been paid in full; 1.3)'Update' shall refer to maintenance releases specific to the License, as they will be made by
I accept the terms of the License Agreement
I do NOT accept the terms of the License Agreement
InstallAnywhere by Zero G
Cancel
Previous
Next

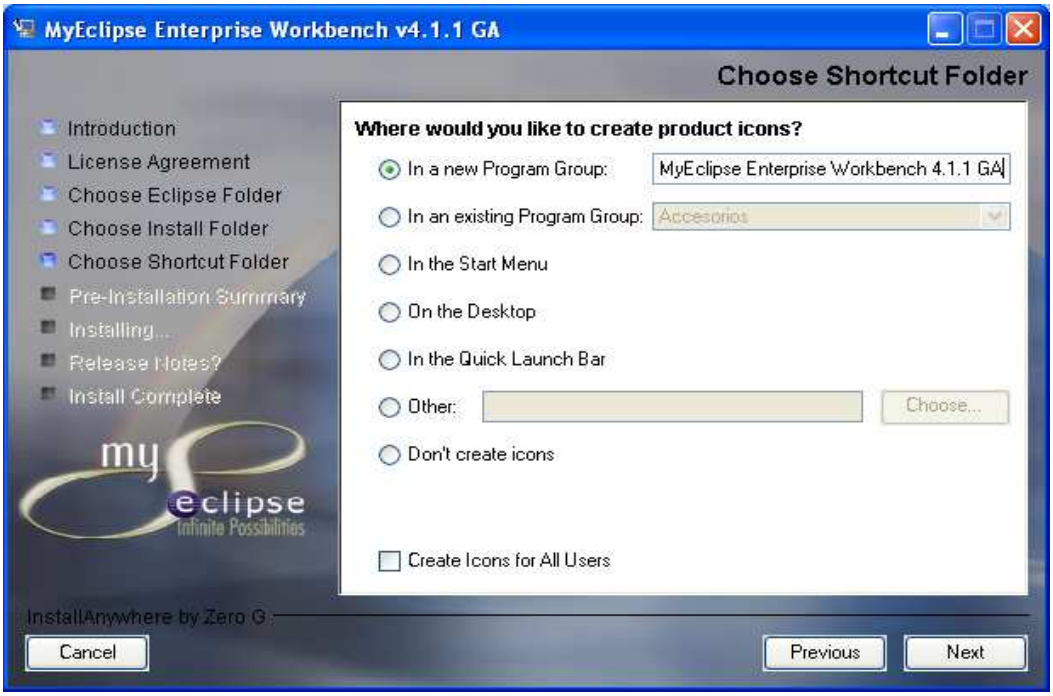
4. Escogemos la ruta en la que se encuentra instalado el IDE Eclipse, es este caso C:\bin\eclipse. Presionamos el botón Next, para obtener la siguiente pantalla.



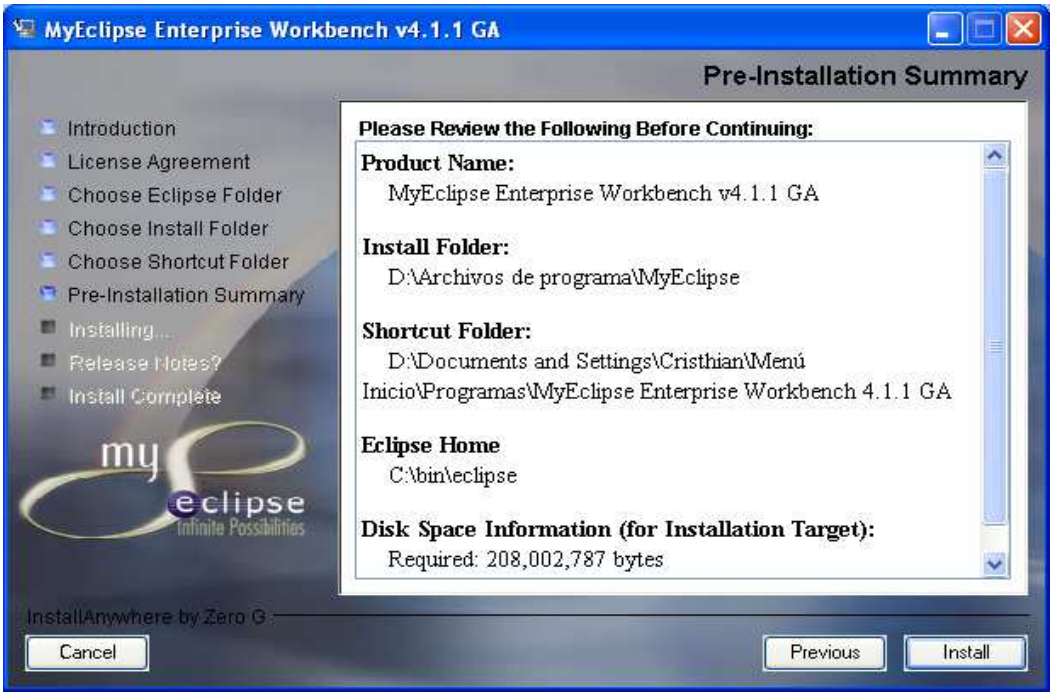
5. Escogemos la ruta en la que desea instalar el IDE MyEclipse, es este caso dejamos seleccionada la opción por defecto D:\Archivos de programa\MyEclipse. Presionamos el botón Next, para obtener la siguiente pantalla.


MyEclipse Enterprise Workbench v4.1.1 GA
Choose Install Folder
Introduction
License Agreement
Choose Eclipse Folder
Choose Install Folder
Choose Shortcut Folder
Pre-Installation Summary
Installing...
Release Notes?
Install Complete
MyEclipse Enterprise Workbench v4.1.1 GA can be installed anywhere on your file system and should NOT be installed underneath C:\bin\eclipse. In order to avoid any potential conflicts, it is highly recommended that you specify a directory that does not already exist.
Where Would You Like to Install MyEclipse Enterprise Workbench ...
D:\Archivos de programa\MyEclipse
Restore Default Folder
Choose...
InstallAnywhere by Zero G
Cancel
Previous
Next

6. Escogemos la opción de en donde deseamos crear iconos del producto MyEclipse, en este caso dejamos seleccionada la opción por defecto In a new Program Group. Presionamos el botón Next, para obtener la siguiente pantalla.

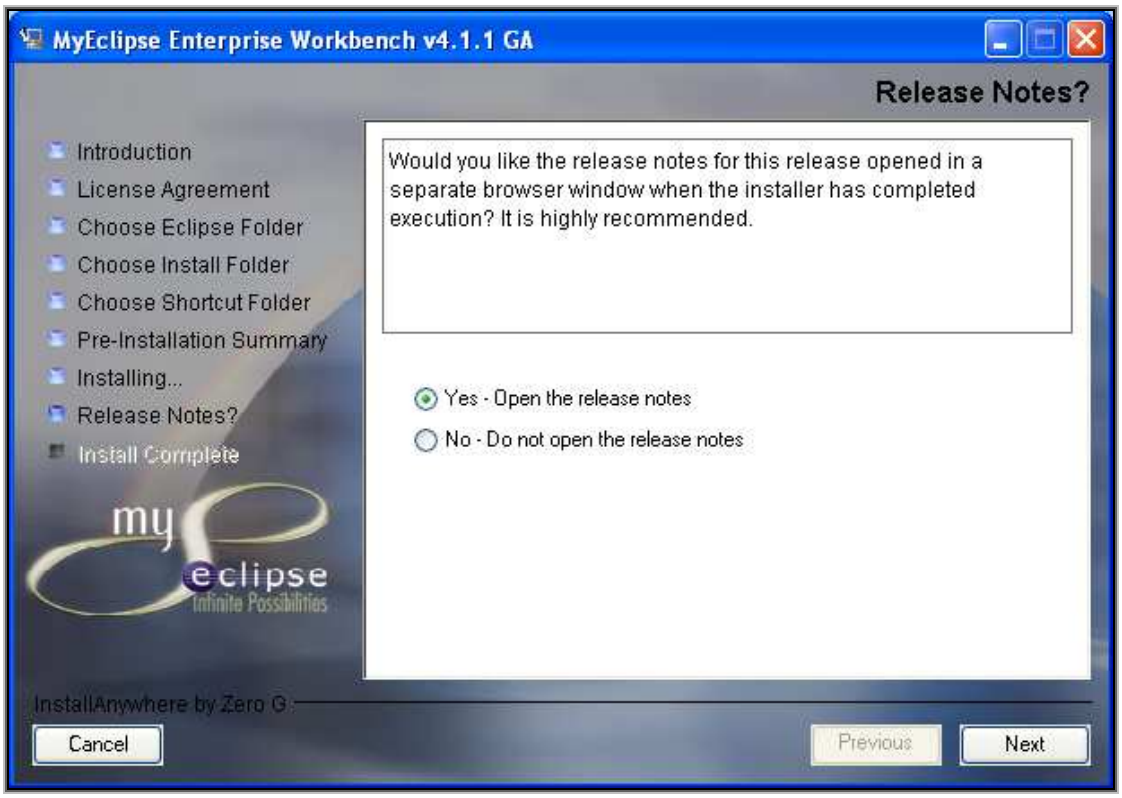


7. Se presenta una pantalla en la que debemos verificar la ruta y espacio de las opciones seleccionadas anteriormente, para proceder a la instalación. Presionamos el botón Install, para obtener la siguiente pantalla.

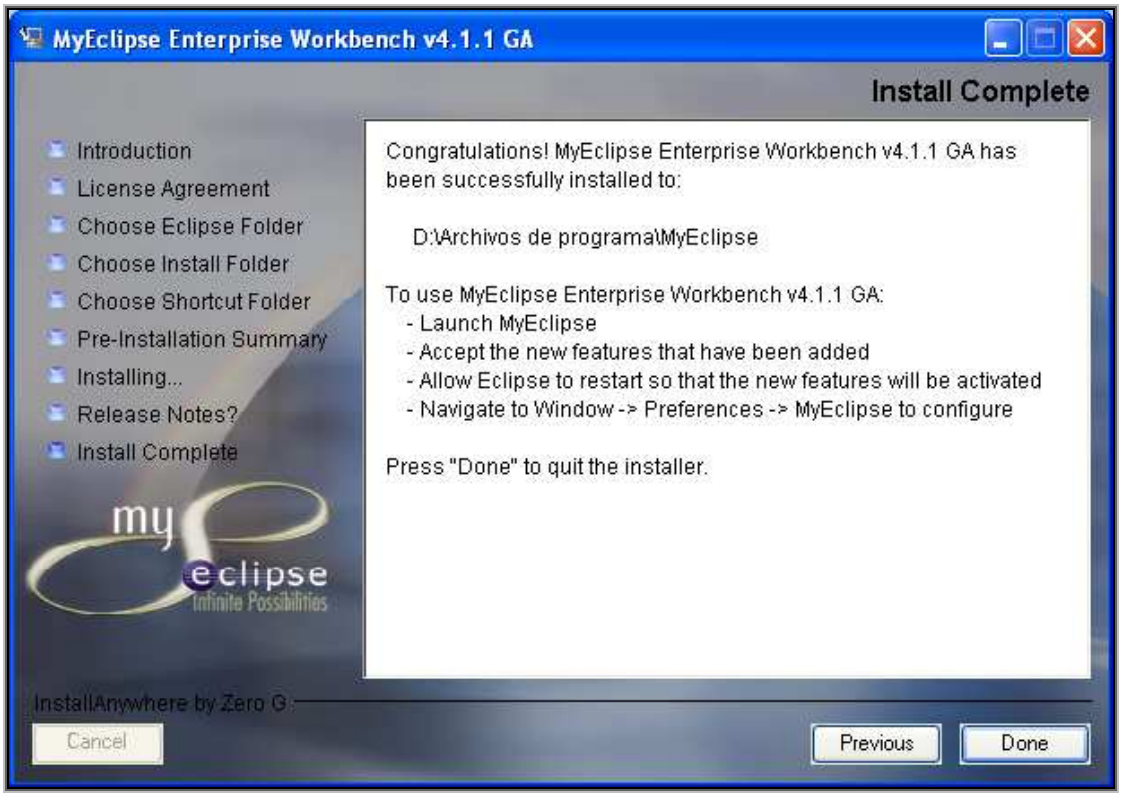


8. Se instalará el IDE MyEclipse.

9. Seleccionamos la opción de abrir para leer el archivo que posee las características del IDE MyEclipse. Presionamos el botón Next para obtener la siguiente pantalla.


MyEclipse Enterprise Workbench v4.1.1 GA
Release Notes?
Introduction
License Agreement
Choose Eclipse Folder
Choose Install Folder
Choose Shortcut Folder
Pre-Installation Summary
Installing...
Release Notes?
Install Complete
Would you like the release notes for this release opened in a separate browser window when the installer has completed execution? It is highly recommended.
Yes - Open the release notes
No - Do not open the release notes
InstallAnywhere by Zero G
Cancel
Previous
Next

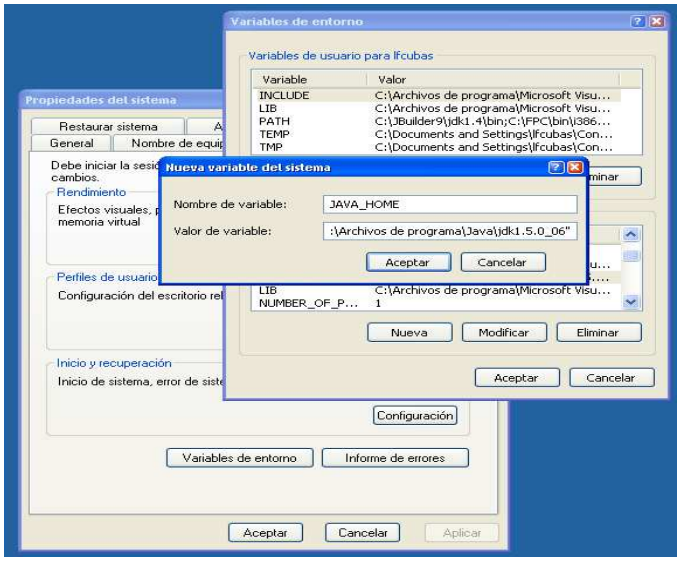
10. Luego de la instalación del IDE MyEclipse aparecerá una ventana indicando que el proceso de instalación ha finalizado. Luego presione el botón Done.


MyEclipse Enterprise Workbench v4.1.1 GA
Install Complete
Introduction
License Agreement
Choose Eclipse Folder
Choose Install Folder
Choose Shortcut Folder
Pre-Installation Summary
Installing...
Release Notes?
Install Complete
Congratulations! MyEclipse Enterprise Workbench v4.1.1 GA has been successfully installed to:
D:\Archivos de programa\MyEclipse
To use MyEclipse Enterprise Workbench v4.1.1 GA:
- Launch MyEclipse
- Accept the new features that have been added
- Allow Eclipse to restart so that the new features will be activated
- Navigate to Window -> Preferences -> MyEclipse to configure
Press "Done" to quit the installer.
my eclipse
Infinite Possibilities
InstallAnywhere by Zero G
Cancel
Previous
Done

11. Con esto finaliza la instalación del servidor IDE MyEclipse.

1.5. Instalación Servidor de Aplicaciones Tomcat 5.5.

Antes de instalar, es muy importante configurar la variable de entorno JAVA_HOME. Tomcat la necesita para funcionar. Hay que ir al Panel de Control, dar click en el icono de Sistema y allí, en la pestaña "Opciones avanzadas", pulsar el botón de "variables de entorno". Aparecerá el diálogo de variables de entorno. La mitad superior está dedicada a las variables de entorno del usuario actual y la inferior a las variables del sistema. Tenemos que escoger crear una nueva variable del sistema:



La nueva variable debe llamarse JAVA_HOME y le pondremos el valor que anotamos antes para el JDK 1.5. En nuestro caso:

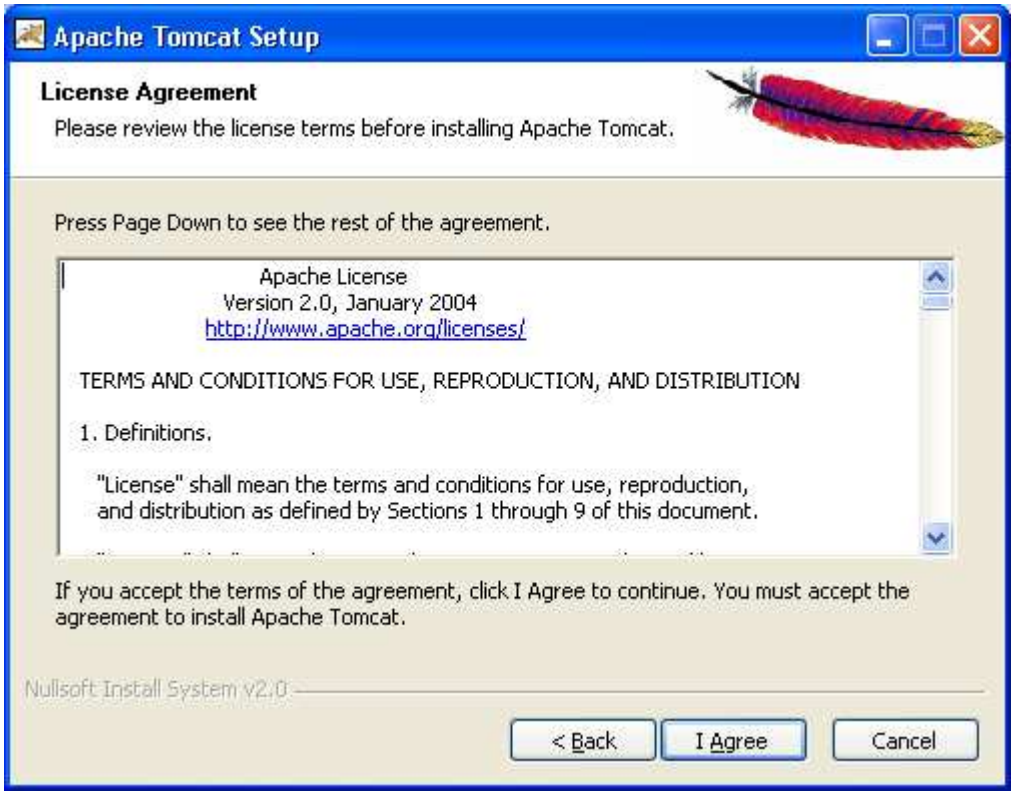
"C:\Archivos de programa\Java\jdk1.5.0_06"

Las comillas son necesarias debido al espacio en blanco en el nombre del directorio.

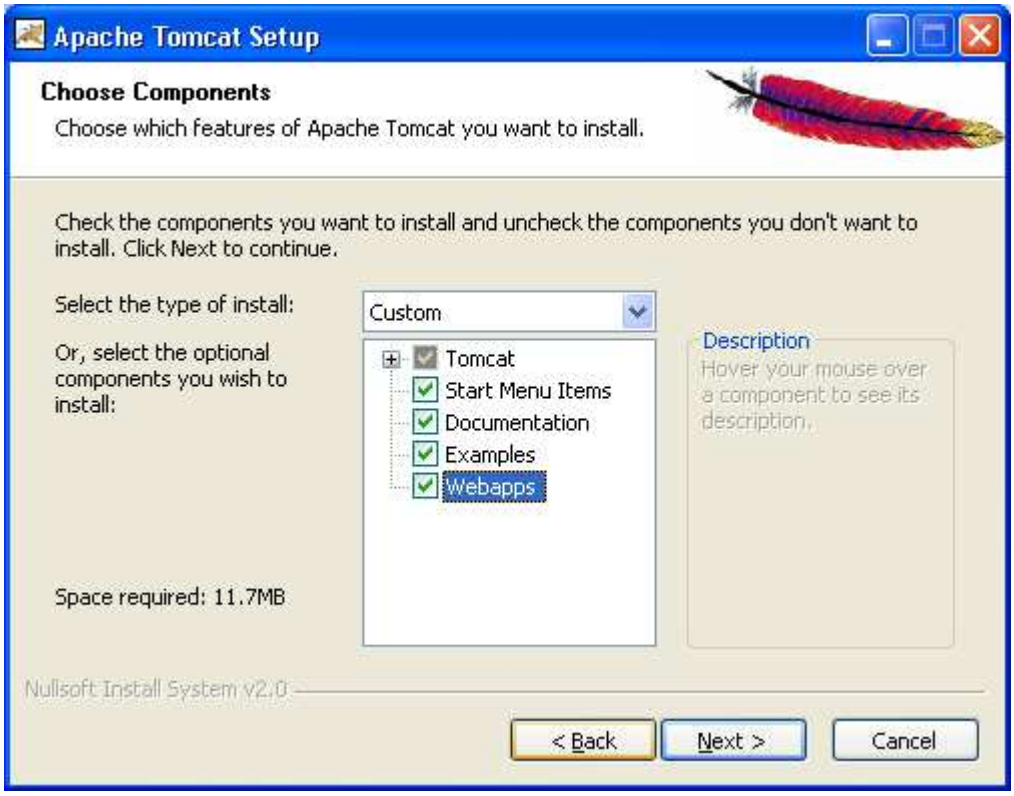
Esta pantalla le da la bienvenida al Asistente de instalación. Haga Click en siguiente.

Apache Tomcat Setup
The Apache Jakarta Project
http://jakarta.apache.org/
Welcome to the Apache Tomcat Setup Wizard
This wizard will guide you through the installation of Apache Tomcat.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel

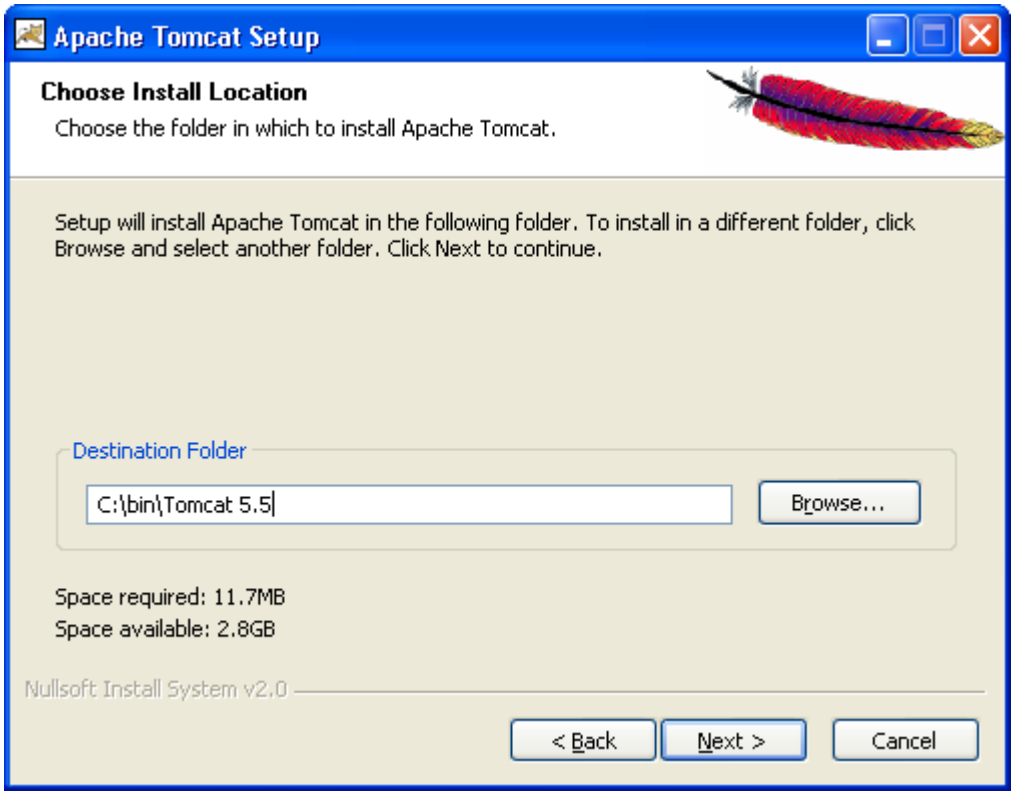
Esta pantalla le indica que debe aceptar las condiciones de uso del software. Haga click en botón Aceptar.


Apache Tomcat Setup
License Agreement
Please review the license terms before installing Apache Tomcat.
Press Page Down to see the rest of the agreement.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.
"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Apache Tomcat.
Nullsoft Install System v2.0
< Back
I Agree
Cancel

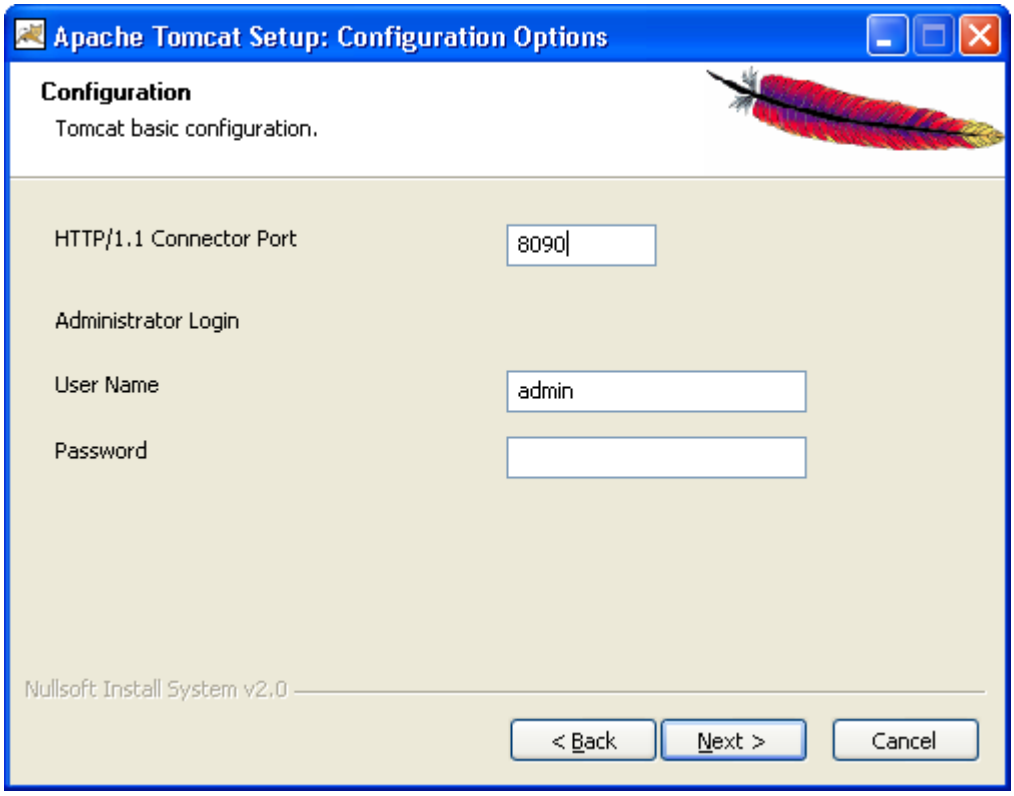
Esta pantalla muestra los componentes que desea instalar y el respectivo espacio requerido. Para aceptar click en el botón siguiente.


Apache Tomcat Setup
Choose Components
Choose which features of Apache Tomcat you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select the type of install:
Custom
Or, select the optional components you wish to install:
Tomcat
Start Menu Items
Documentation
Examples
Webapps
Description
Hover your mouse over a component to see its description.
Space required: 11.7MB
Nullsoft Install System v2.0
< Back
Next >
Cancel

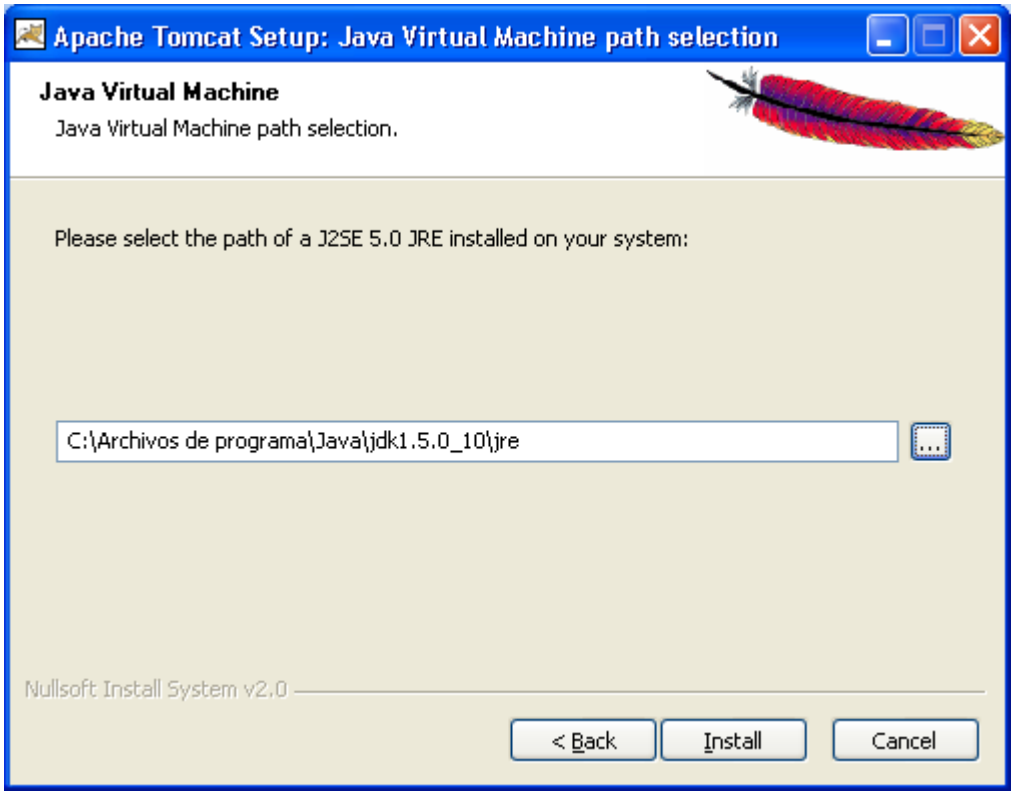
Esta pantalla indica el directorio donde se va a instalar el servidor web. Para una mejor organización debe crear la carpeta bin en la unidad C. Por ejemplo: C:\bin.


Apache Tomcat Setup
Choose Install Location
Choose the folder in which to install Apache Tomcat.
Setup will install Apache Tomcat in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\bin\Tomcat 5.5
Browse...
Space required: 11.7MB
Space available: 2.8GB
Nullsoft Install System v2.0
< Back
Next >
Cancel

La siguiente pantalla indica el ingreso de usuario y clave para el administrador. Además se debe registrar el puerto de conexión. Se recomienda poner el puerto 8090 para un proceso rápido ya que el puerto 8080 es muy solicitado.

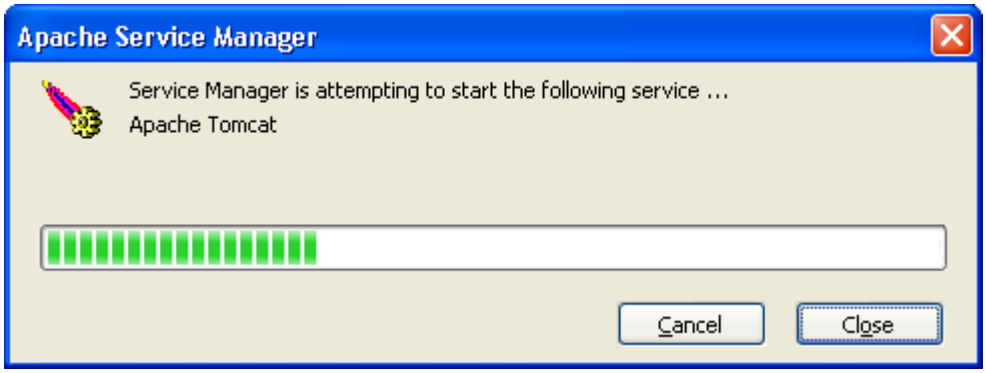


En esta pantalla usted debe indicar la ruta donde reside la maquina virtual de java. C:\Archivos de programa\Java\jdk1.5.0\jre. En este caso se la instalo en la unidad C.


Apache Tomcat Setup: Java Virtual Machine path selection
Java Virtual Machine
Java Virtual Machine path selection.
Please select the path of a J2SE 5.0 JRE installed on your system:
C:\Archivos de programa\Java\jdk1.5.0_10\jre
...
Nullsoft Install System v2.0
< Back
Install
Cancel

La pantalla final de la instalación. Haga clic en el botón finalizar.

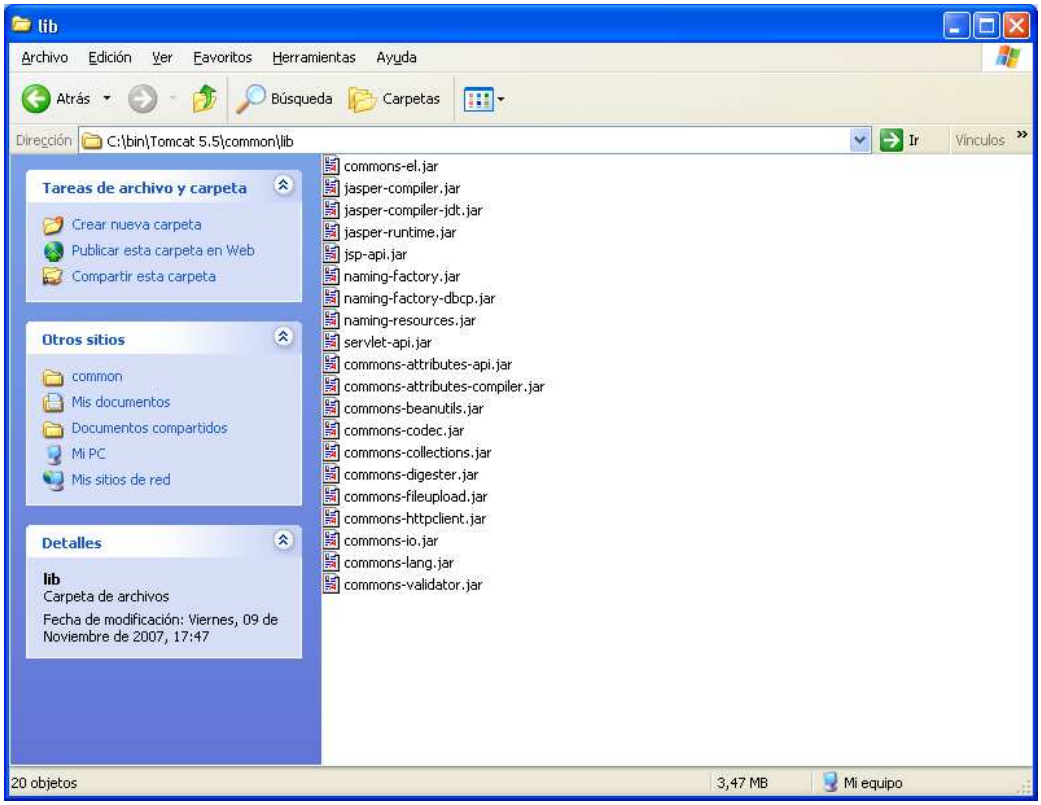
Apache Tomcat Setup
The Apache Jakarta Project
http://jakarta.apache.org/
Completing the Apache Tomcat Setup Wizard
Apache Tomcat has been installed on your computer.
Click Finish to close this wizard.
Run Apache Tomcat
Show Readme
< Back
Finish
Cancel


Apache Service Manager
Service Manager is attempting to start the following service ...
Apache Tomcat
Cancel
Close

1.6. Configuración Del Entorno De Desarrollo Integrado Eclipse

Para ***Configurar*** el tomcat con My eclipse 5.5 se realizaron los siguientes pasos.

1.- Descomprimir el archivo que contiene las librerías para trabajar con aplicaciones Web, este archivo es common-lib.jar. Estos *jar* los copiamos en el directorio donde se encuentra instalado el servidor de aplicaciones en mi caso C:\bin\Tomcat 5.5\common\lib.



2.- Se debe buscar y copiar el jar ***ojdbc14.jar*** de conexión a la base de datos Oracle9i, que se encuentra en el directorio donde está instalado oracle en este caso.

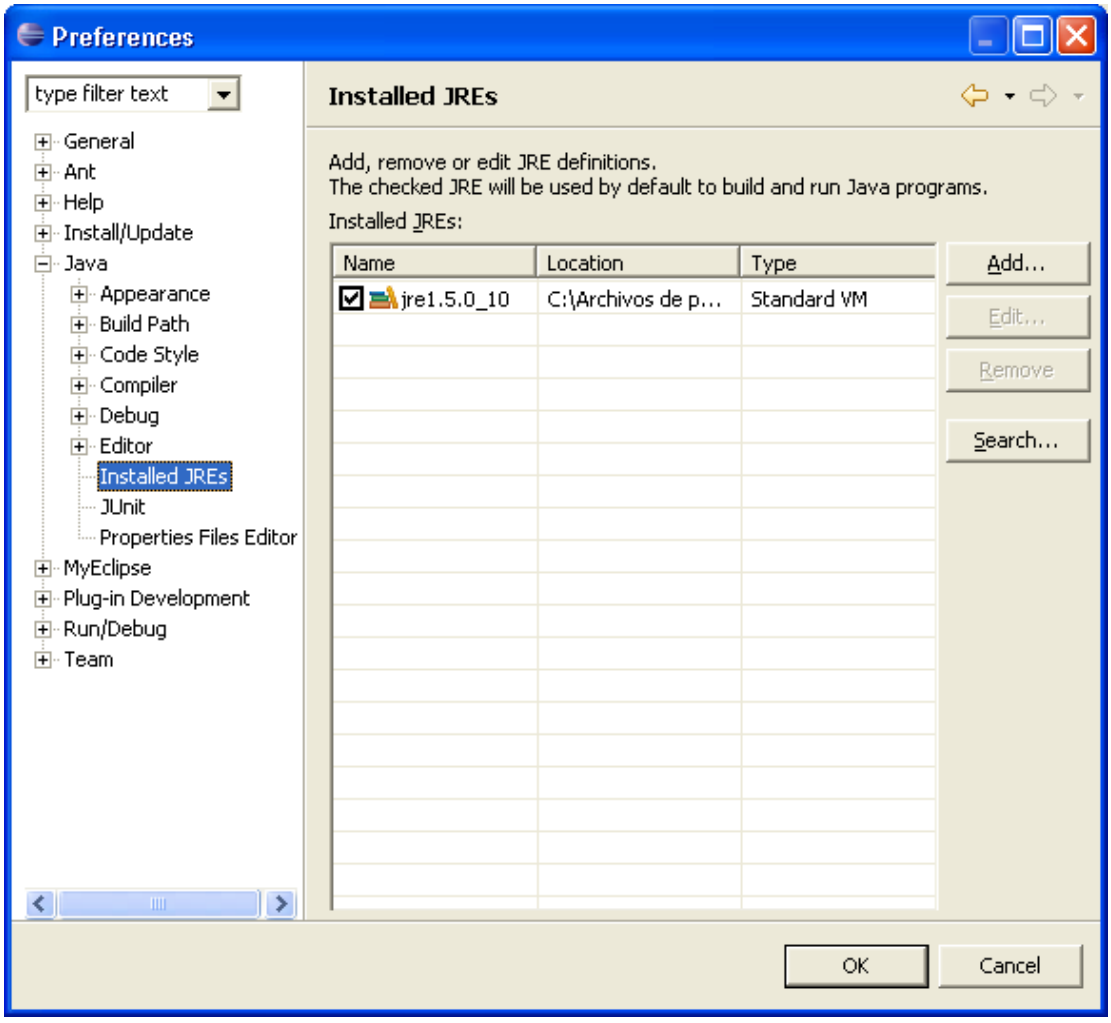
E:\oracle\ora92**jdbc**\lib.

Luego se debe pegar este archivo en el directorio C:\bin\Tomcat 5.5**shared**\lib.

Configuración de Eclipse, JVM, Tomcat 5.5

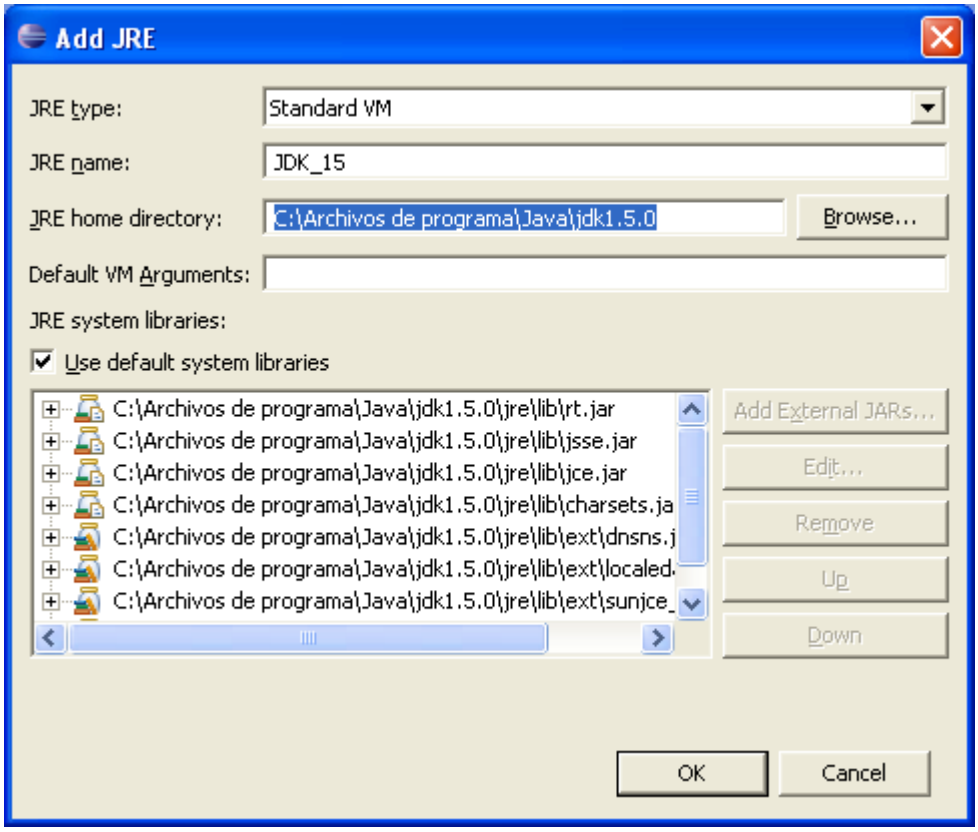
1.- Añadir el kit de java el cual va a utilizar el Servidor de Aplicaiones Tomcat, en este caso realizar lo siguiente.

Click en Windows -> Preferences -> Java -> Installed JREs ->



->Add -> En el botón Browser colocar la ruta de la maquina virtual.

C:\Archivos de programa\Java\jdk1.5.0 renombrarlo **JDK_15.**



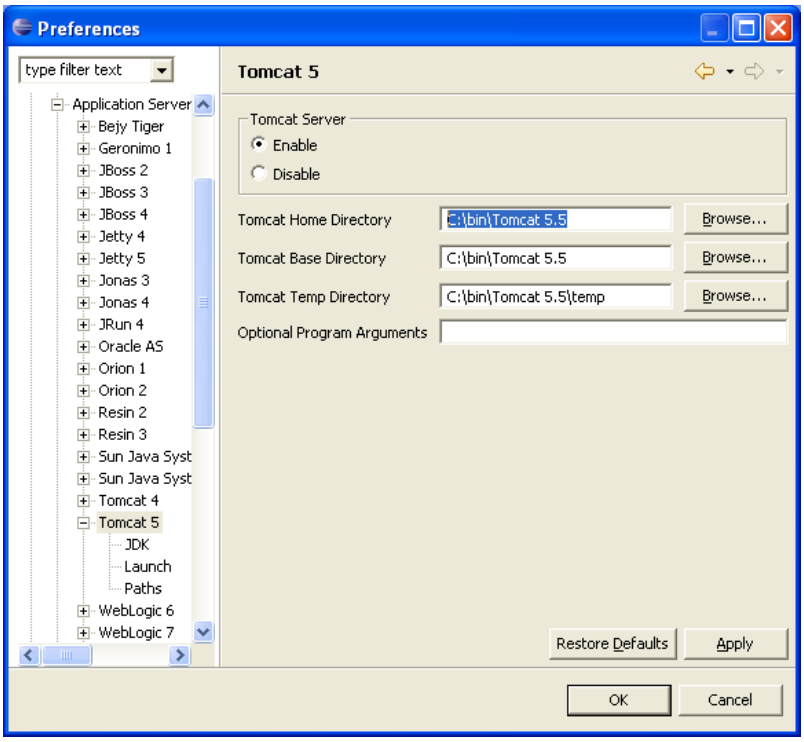
2.- Después de ingresar el nombre deseado en este caso **JDK_15** deben hacer clic en el botón OK.

Luego añadir el Servidor de Aplicaciones Tomcat 5.5 al entorno de eclipse son los siguientes pasos:

3. –Hacer click en el Windows -> Preferences ->> My Eclipse continuar con -> Aplicación Server, seleccionar Tomcat como servidor de aplicaciones.

Después hacer clic en el botón examinar, seleccione el directorio del tomcat C:\bin\Tomcat 5.5 y en el tercer directorio poner al final *Temp.*

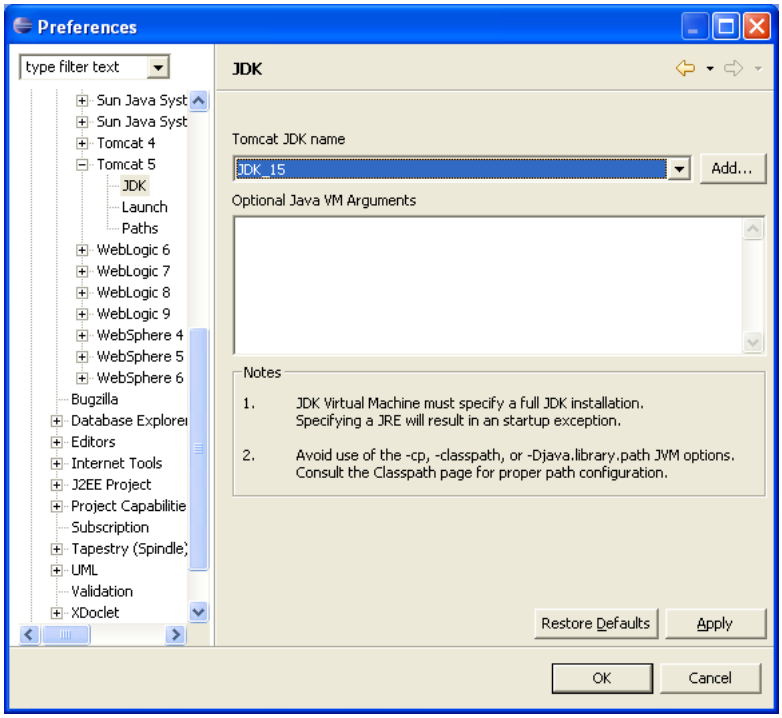
No olvidar escoger la opción **Enable**.



Para establecer el kit de java con el cual va a trabajar el Servidor de Aplicaciones Tomcat 5.5.

4.-Hacer click en el menú Windows -> Preferences ->> MyEclipse -> seleccionar Application Server -> Tomcat -> seleccionar el subárbol

JDK por que trabajaremos con la maquina virtual deseada en este caso escoger JDK_15 (El que creamos en la configuración de la maquina virtual de java). Para aceptar click en el botón *OK.*



1.7. Script de creación de Tablas

Nombre de la tabla: CYP_COSTOS_PROYECTO

Propósito: En esta tabla, se almacenarán los costos del proyecto, siempre contendrá la información de la última consulta que realice el usuario

```
Create table CYP_COSTOS_PROYECTO
(
  CODIGO_PROYECTO              NUMBER not null,
  COSTO_ESTIMADO               NUMBER not null,
  COSTO_REAL                   NUMBER not null,
  FECHA_ULTIMA_CONSULTA        DATE not null,
  CODIGO_PRESUPUESTO           NUMBER not null
)
```

Nombre de la tabla: CYP_DETALLE_COSTOS_PROYECTO.

Propósito: Esta tabla contendrá un histórico de todas las consultas realizadas por el usuario

```
Create table CYP_DETALLE_COSTOS_PROYECTO
(
  CODIGO_DETALLE_COSTO         NUMBER not null,
  CODIGO_COSTO                 NUMBER not null,
  CODIGO_USUARIO               VARCHAR2 (15) not null,
  FECHA_CONSULTA               DATE not null,
  COSTO_REAL                   NUMBER not null,
  CODIGO_PRESUPUESTO           NUMBER
)
```

Nombre de la tabla: CYP_MOVIMIENTO_PRESUPUESTO.

Propósito: Esta es una tabla transaccional en la que se van a registrar todos los movimientos que tenga el presupuesto anual.

```
create table CYP_MOVIMIENTO_PRESUPUESTO
(
  CODIGO_MOVIMIENTO     NUMBER not null,
  CODIGO_PROYECTO       NUMBER not null,
  CODIGO_PRESUPUESTO    NUMBER not null,
  VALOR                 NUMBER not null,
  FECHA_MOVIMIENTO      DATE not null,
  CODIGO_USUARIO        VARCHAR2 (15) not null,
  TIPO_MOVIMIENTO       VARCHAR2(1) not null,
  ESTADO                VARCHAR2 (1)
)
```

Nombre de la tabla: CYP_PRESUPUESTO.

Propósito: En esta tabla se guarda el presupuesto todos los años, de cada uno de los tipos de presupuesto.

```
Create table CYP_PRESUPUESTO
(
  CODIGO_PRESUPUESTO          NUMBER not null,
  CODIGO_TIPO_PRESUPUESTO     VARCHAR2 (3) not null,
  ANIO                        VARCHAR2 (5) not null,
  ESTADO                      VARCHAR2 (1) not null,
  VALOR_INICIAL               NUMBER not null,
  SALDO                       NUMBER not null,
  VALOR_FINAL                 NUMBER not null,
  APORTE_ANIO_ANT             NUMBER
)
```

Nombre de la tabla: CYP_TIPOS_PRESUPUESTOS.

Propósito: Esta tabla contiene los tipos de presupuestos que posee el área como por ejemplo: Presupuesto de Hardware, Presupuesto de Software, Presupuesto para Recursos Humanos, etc.

```
Create table CYP_TIPOS_PRESUPUESTOS
(
  CODIGO_TIPO_PRESUPUESTO   VARCHAR2 (3) not null,
  NOMBRE                    VARCHAR2 (50) not null,
  ESTADO                    VARCHAR2 (1) not null,
  FECHA_INICIO              DATE not null,
  FECHA_FIN                 DATE
)
```

Nombre de la tabla: CYP_TRANSFERENCIAS

Propósito: Esta es una tabla Transaccional, en la que se van a registrar las transferencias de presupuestos y transferencias entre un área de la empresa y Sistemas, por lo que interactúa con la tabla CYP_PRESUPUESTO

```
Create table CYP_TRANSFERENCIAS
(
  CODIGO_TRANSFERENCIA      NUMBER not null,
  TIPO_TRANSFERENCIA        VARCHAR2 (1) not null,
  CODIGO_PRESUPUESTO        NUMBER not null,
  CODIGO_USUARIO            VARCHAR2 (15) not null,
  VALOR                     NUMBER not null,
  ESTADO                    VARCHAR2 (1) not null,
  FECHA_INICIO              DATE not null,
  FECHA_FIN                 DATE,
  OBSERVACION               VARCHAR2 (100),
  CODIGO_AREA               NUMBER (3) not null,
  PRESUPUESTO_ORIGEN        NUMBER
)
```

Nombre de la tabla: CYP_COSTOS_RECURSOS.

Propósito: Esta tabla contiene los costos promedios de los recursos que intervienen en un determinado proyecto como por ejemplo: costos de Hardware, costos de Software, costos para Recursos Humanos, etc.

```
Create table CYP_COSTOS_RECURSOS
(
  CODIGO_COSTO_RECURSO       NUMBER not null,
  CODIGO_TIPO_PARTICIPANTE   VARCHAR2(10),
  VALOR                      NUMBER not null,
  ESTADO                     VARCHAR2 (1) not null,
  FECHA_INICIO               DATE not null,
  FECHA_FIN                  DATE,
  CODIGO_SUB_TIPO_SERVICIO   VARCHAR2 (10)
)
```

1.8. PAQUETES CON FUNCIONES Y PROCEDIMIENTOS ESPECÍFICAS

Nombre*:* P_REGISTRA_PROXIMO_PRESUPESTO

Origen: CYPK_TRX_PRESUPUESTOS

Parámetros: Pv_TramaPresupuesto, Pv_Anio, Pv_Error.

Definición:. Registrar la distribución del presupuesto del área de sistemas para el próximo año

```
procedure P_REGISTRA_PROXIMO_PRESUPESTO  (Pv_TramaPresupuesto     in
varchar2
                                ,Pv_Anio              in     varchar2
                                ,Pv_Error               out varchar2) is

 ---------------------------------------------------------------------------------
 -- Purpose   : Registrar la distribución del presupuesto del área de sistemas para el
 --               próximo anio
 -- Author     : Grupo # 9
 -- Created    : 10/08/2007
 -- Parameter : Pn_Presupuesto_Sistemas Ingresa: el valor total del presupuesto
 --               lv_trama_presupuesto    Ingresa: el valor de cada tipo de resupuesto
 --               Pv_Error              Retorna: el mensaje de error, si ocurriese alguno
 ---------------------------------------------------------------------------------


  --Consulta si el tipo de presupuesto que se desea registrar existe
  cursor C_TipoPresupuesto (Cv_Tipo_Presupuesto   varchar2) is
  select tp.nombre
  from   cyp_tipos_presupuestos tp
  where  tp.codigo_tipo_presupuesto = Cv_Tipo_Presupuesto
  and    tp.estado = 'A';

  --Verifica si ya se ha registrado el presupuesto para el año que se intenta guardar
  cursor C_ExistePresupuesto (Cv_Tipo_Presupuesto   varchar2) is
  select p.codigo_presupuesto,
         p.valor_inicial,
         p.valor_final,
         p.saldo
  from   cyp_presupuesto p
  where  p.codigo_tipo_presupuesto = Cv_Tipo_Presupuesto
  and    p.anio = Pv_Anio;
```

```
   --Para verificar que el anio que se desea registrar es el próximo
   cursor C_VerificaAnio is
   select to_char(to_number(to_char(sysdate,'yyyy'))+1)
   from   dual;

   Lc_ExistePresupuesto      C_ExistePresupuesto%rowtype;

   Lv_NombrePrograma         varchar2(60);
   Lv_Caracter               varchar2(1);
   Lv_Palabra                varchar2(20);
   Lv_TipoPresupuesto        varchar2(3);
   Lv_NombrePresupuesto      varchar2(30);
   Lv_Estado                 varchar2(1);
   Lv_ProximoAnio            varchar2(4);

   Ln_TamañoTrama            number;
   Ln_ValorPresupuesto       number;
   Ln_Contador               number:=0;
   Ln_TotalPresupuesto       number:=0;
   Ln_CodigoPresupuesto      number;
   Ln_SaldoPresupuesto       number;
   Ln_ValorInicial           number;
   Ln_ValorFinal             number;
   Ln_PresupuestoSistemas    number;

   Lb_Found                  boolean;

   Le_Error                  exception;

  begin

   Lv_NombrePrograma :=
'CYPK_TRX_PRESUPUESTOS.P_REGISTRA_PROXIMO_PRESUPESTO';

   --Obtener el tamaño de la Trama que contiene el detalle del presupuesto
   Ln_TamañoTrama := length(Pv_TramaPresupuesto);

   --Verificar Anio
   open C_VerificaAnio;
   fetch C_VerificaAnio into Lv_ProximoAnio;
   close C_VerificaAnio;

   if Lv_ProximoAnio <> Pv_Anio then
     Pv_Error := 'Solo puede registrar el presupuesto para el próximo Anio';
     raise Le_Error;
   end if;

   --Descomponer trama
```

```
    for i in 1..Ln_TamañoTrama loop
      Lv_Caracter := substr(Pv_TramaPresupuesto,i,1);
      if Lv_Caracter != '|' then
        Lv_Palabra := Lv_Palabra||Lv_Caracter;
      else
        Ln_Contador := Ln_Contador+1;
        if Ln_Contador = 1 then
          Lv_TipoPresupuesto := Lv_Palabra;

          --Verifica si el tipo de presupuesto que se desea registrar es válido
          open C_TipoPresupuesto(Lv_TipoPresupuesto);
          fetch C_TipoPresupuesto into Lv_NombrePresupuesto;
          Lb_Found := C_TipoPresupuesto%notfound;
          close C_TipoPresupuesto;

          if Lb_Found then
            Pv_Error := 'El tipo de presupuesto que desea registrar no es válido';
            raise Le_Error;
          end if;

          Lv_Palabra := null;

        else

          Ln_ValorPresupuesto := to_number(replace(Lv_Palabra,'.',','));

          --Si el tipo de presupuesto es el que indica el valor total
          if Lv_TipoPresupuesto = 'PSI' then
            --Siempre la primera posición de la cadena es para el presupuesto Total de
Sistemas
            Ln_PresupuestoSistemas := Ln_ValorPresupuesto;
          else
            Ln_TotalPresupuesto := Ln_TotalPresupuesto + Ln_ValorPresupuesto;
          end if;

          --Si la suma de los tipos de presupuesto es mayor al total
          if Ln_TotalPresupuesto > Ln_PresupuestoSistemas then
            Pv_Error := 'La suma de los Tipos de Presupuesto no puede ser mayor al
Total del presupuesto de Sistemas que es $ '||Ln_PresupuestoSistemas;
            raise Le_Error;
          end if;

          --Obtengo los valores
          Ln_ValorInicial := Ln_ValorPresupuesto;
          Ln_SaldoPresupuesto := Ln_ValorPresupuesto;
          Ln_ValorFinal := Ln_ValorPresupuesto;

          --Verificar si el Tipo de Presupuesto ya esta registrado para el año indicado
```

```
        open C_ExistePresupuesto (Lv_TipoPresupuesto);
        fetch C_ExistePresupuesto into Lc_ExistePresupuesto;
        Lb_Found := C_ExistePresupuesto%found;
        close C_ExistePresupuesto;

        if Lb_Found then --Existe

          --Actualizar valores a los tipos de presupuesto
          CYPK_OBJ_PRESUPUESTOS.P_UPD_PRESUPUESTO
(Pn_CodigoPresupuesto  => Lc_ExistePresupuesto.Codigo_Presupuesto
                                    ,Pn_ValorInicial       => Ln_ValorInicial
                                    ,Pn_saldo              => Ln_SaldoPresupuesto
                                    ,Pn_ValorFinal         => Ln_ValorFinal
                                    ,Pn_TransferenciaAreas => null
                                    ,Pv_Error              => Pv_Error);

          if Pv_Error is not null then
            raise Le_Error;
          end if;
        else --No Existe

          Lv_Estado := 'P';

          --insertar el valor del tipo de presupuesto
          CYPK_OBJ_PRESUPUESTOS.P_INS_PRESUPUESTO
(Pn_CodigoTipoPres => Lv_TipoPresupuesto
                                    ,Pv_Año            => pv_anio
                                    ,Pv_Estado         => Lv_Estado
                                    ,Pn_ValorInicial   => Ln_ValorInicial
                                    ,Pn_saldo          => Ln_SaldoPresupuesto
                                    ,Pn_ValorFinal     => Ln_ValorFinal
                                    ,Pn_AporteAnioAnt  => 0
                                    ,Pn_CodigoPresup   => Ln_CodigoPresupuesto
                                    ,Pv_Error          => Pv_Error);

          if Pv_Error is not null then
            raise Le_Error;
          end if;
        end if;

        Ln_Contador := 0;
        Lv_Palabra  := null;

      end if;
    end if;
  end loop;

  --Si la suma de los tipos de presupuesots es menor que el total
```

```
    if Ln_TotalPresupuesto < Ln_PresupuestoSistemas then
      Pv_Error := 'La suma de los Tipos de Presupuestos da $
'||Ln_TotalPresupuesto||', pero debe ser igual al Valor Total del presupuesto de
Sistemas que es $ '||Ln_PresupuestoSistemas;
      raise Le_Error;
    end if;

    commit;

  exception
    when Le_Error then
      Pv_Error := 'Error: '|| Pv_Error;
      rollback;
    when others then
      pv_error := Lv_NombrePrograma||' - '||sqlerrm;
      rollback;

  end P_REGISTRA_PROXIMO_PRESUPESTO;
```

Nombre*:* P_TRANSFERENCIA_AREAS

Origen: CYPK_TRX_PRESUPUESTOS

Parámetros:Pn_CodigoTransferencia ,Pv_TipoTransferencia, Pv_CodigoUsuario, Pn_Valor, Pv_Observacion, Pn_CodigoArea, Pv_TramaTransferencia, Pv_Error

Definición: Registra transferencias de presupuesto entre Sistemas y otras áreas.

```
  procedure P_TRANSFERENCIA_AREAS          (Pn_CodigoTransferencia   in
number
                                ,Pv_TipoTransferencia  in     varchar2
                                ,Pv_CodigoUsuario       in     varchar2
                                ,Pn_Valor                  in     number
                                ,Pv_Observacion        in     varchar2
                                ,Pn_CodigoArea         in     number
                                ,Pv_TramaTransferencia    in     varchar2
                                ,Pv_Error                     out varchar2) is
```

```
-------------------------------------------------------------------------------------
-- Purpose   :
-- Author    : Grupo # 9
-- Created   : 10/08/2007
-- Parameter : Pv_Error          Retorna: el mensaje de error, si ocurriese alguno
-------------------------------------------------------------------------------------

 --Consulta si el tipo de presupuesto que se desea registrar existe
  cursor C_TipoPresupuesto (Cv_Tipo_Presupuesto   varchar2) is
  select tp.nombre
  from   cyp_tipos_presupuestos tp
  where  tp.codigo_tipo_presupuesto = Cv_Tipo_Presupuesto
  and    tp.estado = 'A';


  --Verifica si ya se ha registrado el presupuesto para el año que se intenta guardar
  cursor C_ExistePresupuesto (Cv_Tipo_Presupuesto   varchar2) is
  select p.codigo_presupuesto,
         p.valor_inicial,
         p.valor_final,
         p.saldo,
         p.transferencia_areas
  from   cyp_presupuesto p
  where  p.codigo_tipo_presupuesto = Cv_Tipo_Presupuesto
  and    p.anio = to_char(sysdate,'yyyy');



  Lc_ExistePresupuesto      C_ExistePresupuesto%rowtype;

  Lv_NombrePrograma         varchar2(60);
  Lv_Caracter               varchar2(1);
  Lv_Palabra                varchar2(20);
  Lv_TipoPresupuesto        varchar2(3);
  Lv_NombrePresupuesto      varchar2(30);

  Ln_TamañoTrama            number;
  Ln_ValorPresupuesto       number;
  Ln_Contador               number:=0;
  Ln_TotalPresupuesto       number:=0;
  Ln_SaldoPresupuesto       number;
  Ln_ValorInicial           number;
  Ln_ValorFinal             number;
  Ln_Verifica_total         number:=0;

  Lb_Found                  boolean;

  Le_Error                  exception;
```

```
  begin

    Lv_NombrePrograma :=
'CYPK_TRX_PRESUPUESTOS.P_TRANSFERENCIA_AREAS';

    --Obtener el tamaño de la Trama que contiene el detalle del presupuesto
    Ln_TamañoTrama := length(Pv_TramaTransferencia);

    --Verificar si el Tipo de Presupuesto ya esta registrado para el año indicado
    open C_ExistePresupuesto ('PSI');
    fetch C_ExistePresupuesto into Lc_ExistePresupuesto;
    Lb_Found := C_ExistePresupuesto%found;
    close C_ExistePresupuesto;

   --Obtengo los valores
    Ln_ValorInicial := Lc_ExistePresupuesto.Valor_Inicial;
    Ln_SaldoPresupuesto := Lc_ExistePresupuesto.Saldo+Pn_Valor;
    Ln_ValorFinal := Lc_ExistePresupuesto.Valor_Final+Pn_Valor;

    --De aqui saque el insert a transferencias

    --Actualizar valores a los tipos de presupuesto
    CYPK_OBJ_PRESUPUESTOS.P_UPD_PRESUPUESTO
(Pn_CodigoPresupuesto  => Lc_ExistePresupuesto.Codigo_Presupuesto
                         ,Pn_ValorInicial       => Ln_ValorInicial
                         ,Pn_saldo              => Ln_SaldoPresupuesto
                         ,Pn_ValorFinal         => Ln_ValorFinal
                         ,Pn_TransferenciaAreas => Pn_Valor
                         ,Pv_Error              => Pv_Error);

    if Pv_Error is not null then
      raise Le_Error;
    end if;


    --Descomponer trama
    for i in 1..Ln_TamañoTrama loop
      Lv_Caracter := substr(Pv_TramaTransferencia,i,1);
      if Lv_Caracter != '|' then
        Lv_Palabra := Lv_Palabra||Lv_Caracter;
      else
        Ln_Contador := Ln_Contador+1;
        if Ln_Contador = 1 then
          Lv_TipoPresupuesto := Lv_Palabra;

          --Verifica si el tipo de presupuesto que se desea registrar es válido
          open C_TipoPresupuesto(Lv_TipoPresupuesto);
```

```
        fetch C_TipoPresupuesto into Lv_NombrePresupuesto;
        Lb_Found := C_TipoPresupuesto%notfound;
        close C_TipoPresupuesto;

        if Lb_Found then
          Pv_Error := 'El tipo de presupuesto que desea registrar no es válido';
          raise Le_Error;
        end if;

        Lv_Palabra := null;

       else

        Ln_ValorPresupuesto := to_number(replace(Lv_Palabra,'.',','));
        Ln_TotalPresupuesto := Ln_TotalPresupuesto + Ln_ValorPresupuesto;

        --Si la suma de los tipos de presupuesto es mayor al total
        if Ln_TotalPresupuesto > Pn_Valor then
          Pv_Error := 'La suma de los Tipos de Presupuesto no puede ser mayor al
                      Total de la Transferencia '||Pn_Valor;
          raise Le_Error;
        end if;


        --Verificar si el Tipo de Presupuesto ya esta registrado para el año indicado
        open C_ExistePresupuesto (Lv_TipoPresupuesto);
        fetch C_ExistePresupuesto into Lc_ExistePresupuesto;
        Lb_Found := C_ExistePresupuesto%found;
        close C_ExistePresupuesto;

        --Verifico si no excede al total ingresado
        Ln_Verifica_total := Ln_Verifica_total + Ln_ValorPresupuesto;

       --Obtengo los valores
        Ln_ValorInicial := Lc_ExistePresupuesto.Valor_Inicial;
        Ln_SaldoPresupuesto := Lc_ExistePresupuesto.Saldo+Ln_ValorPresupuesto;
        Ln_ValorFinal := Lc_ExistePresupuesto.Valor_Final+Ln_ValorPresupuesto;


        --Aqui agregue el llamado a transferencias
        CYPK_OBJ_PRESUPUESTOS.P_INS_TRANSFERENCIAS
Pn_CodigoTransferencia => Pn_CodigoTransferencia
                                    ,Pv_TipoTransferencia   => Pv_TipoTransferencia
                                    ,Pn_CodigoPresupuesto   =>
Lc_ExistePresupuesto.Codigo_Presupuesto
                                    ,Pv_CodigoUsuario       => Pv_CodigoUsuario
                                    ,Pn_Valor               => Ln_ValorPresupuesto
                                    ,Pv_Estado              => 'I'
```

```
                                    ,Pv_Observacion        => Pv_Observacion
                                    ,Pn_CodigoArea         => Pn_CodigoArea
                                    ,Pv_Error              => Pv_Error);

        if Pv_Error is not null then
          raise Le_Error;
        end if;

        --Actualizar valores a los tipos de presupuesto
        CYPK_OBJ_PRESUPUESTOS.P_UPD_PRESUPUESTO
(Pn_CodigoPresupuesto  => Lc_ExistePresupuesto.Codigo_Presupuesto
                                    ,Pn_ValorInicial       => Ln_ValorInicial
                                    ,Pn_saldo              => Ln_SaldoPresupuesto
                                    ,Pn_ValorFinal         => Ln_ValorFinal
                                    ,Pn_TransferenciaAreas =>
(Lc_ExistePresupuesto.transferencia_areas + Ln_ValorPresupuesto)
                                    ,Pv_Error              => Pv_Error);

        if Pv_Error is not null then
          raise Le_Error;
        end if;

        Ln_Contador := 0;
        Lv_Palabra  := null;

       end if;
     end if;
   end loop;

   if Ln_Verifica_total <> Pn_Valor then
     Pv_Error := 'Los valores distribuidos no suman el monto de la transferecia:
'||Pn_Valor;
     raise Le_Error;
   end if;
      commit;
      Pv_Error := 'Transferencia Realizada con Exito';

  exception
    when Le_Error then
      Pv_Error := Lv_NombrePrograma ||' - '|| Pv_Error;
      rollback;
    when others then
      pv_error := Lv_NombrePrograma||' - '||sqlerrm;
      rollback;

  end P_TRANSFERENCIA_AREAS;
```

Nombre*:* P_TRANSFERENCIA_PROYECTOS

Origen: CYPK_TRX_PRESUPUESTOS

Parámetros: Pv_CodigoProyecto, Pv_CodigoUsuario, Pv_ValorTotal, Pv_Trama Pv_Tipo, Pv_Error

Definición: Se realiza la transferencia de saldos sobrantes de un proyecto finalizado.

```
  procedure P_TRANSFERENCIA_PROYECTOS     (Pv_CodigoProyecto       in
varchar2
                                ,Pv_CodigoUsuario       in     varchar2
                                ,Pv_ValorTotal          in     varchar2
                                ,Pv_Trama               in     varchar2
                                ,Pv_Tipo                in     varchar2
                                ,Pv_Error                 out varchar2) is

 ---------------------------------------------------------------------------------------
 -- Purpose   :
 -- Author    : Grupo # 9
 -- Created   : 10/08/2007
 -- Parameter : Pv_Error          Retorna: el mensaje de error, si ocurriese alguno
 ---------------------------------------------------------------------------------------

   --Saber si existe el proyecto
   cursor C_Proyecto (Cv_Estado   varchar2) is
   select *
   from   gen_proyectos p
   where  p.codigo_proyecto = to_number(Pv_CodigoProyecto)
   and    p.codigo_estado = Cv_Estado;

  --Verificar si existe el tipo de presupuesto y recupera sus respectivos  valores
   cursor C_TipoPresupuesto (Cv_Tipo_Presupuesto   varchar2) is
   select p.codigo_presupuesto,
          p.saldo,
          p.valor_inicial,
          p.valor_final,
          p.transferencia_areas
   from   cyp_presupuesto p
   where  p.codigo_tipo_presupuesto = Cv_Tipo_Presupuesto
   and    p.estado = 'A';
```

```
   cursor C_DatosPsi (Cv_TipoPresupuesto   varchar2) is
   select p.codigo_presupuesto,
          p.saldo,
          p.valor_inicial,
          p.valor_final,
          p.transferencia_areas
   from   cyp_presupuesto p
   where  p.codigo_tipo_presupuesto = Cv_TipoPresupuesto
   and    p.estado = 'A';

   Cursor C_VerificaDevolucion is
   select finalizado
   from   cyp_costos_proyecto cp
   where  cp.codigo_proyecto = to_number(Pv_CodigoProyecto)
   and    cp.finalizado is null;

   Lc_Proyecto             C_Proyecto%rowtype;
   Lc_TipoPresupuesto      C_TipoPresupuesto%rowtype;
   Lc_DatosPsi             C_DatosPsi%rowtype;

   Lv_NombrePrograma       varchar2(50);
   Lv_Caracter             varchar2(1);
   Lv_Palabra              varchar2(20);
   Lv_CodTipoPresupuesto   varchar2(3);
   Lv_TipoMovimiento       varchar2(1);
   Lv_Verifica             varchar2(1);
   Lv_Estado               varchar2(3);

   Ln_Saldo                number;
   Ln_CodigoMovimiento     number;
   Ln_TamañoTrama          number;
   Ln_ValorPresupuesto     number;
   Ln_Contador             number:=0;
   Ln_ValorTotal           number:=0;

   Lb_Found                boolean;

   Le_Error                exception;

 begin

   Lv_NombrePrograma :=
'CYPK_TRX_PRESUPUESTOS.P_TRANSFERENCIA_PROYECTOS';

   if Pv_Tipo = 'D' then --Devolución
     Lv_Estado := 'FI';--'FI';
   elsif Pv_Tipo = 'A' then --Asignación
```

```
      Lv_Estado := 'EC';
    end if;

    open C_Proyecto (Lv_Estado);
    fetch C_Proyecto into Lc_Proyecto;
    Lb_Found := C_Proyecto%notfound;
    close C_Proyecto;

    if Lb_Found then
      Pv_Error := 'El proyecto no existe o aun no esta en un estado en el que no se
puede tranferir dinero'; -- o devolver saldo sobrante
      raise Le_Error;
    end if;

    --Obtener datos de PSI
    open C_DatosPsi ('PSI');
    fetch C_DatosPsi into Lc_DatosPsi;
    close C_DatosPsi;

    if Pv_Tipo = 'D' then --Devolución

      --Verificar si aun no he devuelto el excedente
      open C_VerificaDevolucion;
      fetch C_VerificaDevolucion into Lv_Verifica;
      Lb_Found := C_VerificaDevolucion%notfound;
      close C_VerificaDevolucion;

      if Lb_Found then
        Pv_Error := 'El Proyecto No. '||Pv_CodigoProyecto|| ' ya realizo la devolución de
        excedente';
        raise Le_Error;
      end if;

      Lv_TipoMovimiento := '-';--Por que se resta del presupuesto de un proyecto
      Ln_Saldo :=  Lc_DatosPsi.Saldo + to_number(Pv_ValorTotal);--Se suma al total
      del presupuesto de sistemas

    elsif Pv_Tipo = 'A' then --Asignación

      Lv_TipoMovimiento := '+'; --Por que se suma al presupuesto de un proyecto
      Ln_Saldo := Lc_DatosPsi.Saldo - to_number(Pv_ValorTotal); --Se resta del total
                                              del presupuesto de sistemas

    end if;

    --Registrar los datos de PSI
    CYPK_OBJ_PRESUPUESTOS.P_UPD_PRESUPUESTO
(Pn_CodigoPresupuesto  => Lc_DatosPsi.Codigo_Presupuesto
```

```
                                ,Pn_ValorInicial       => Lc_DatosPsi.Valor_Inicial
                                ,Pn_saldo              => Ln_Saldo
                                ,Pn_ValorFinal         => Lc_DatosPsi.Valor_Final
                                ,Pn_TransferenciaAreas =>
Lc_DatosPsi.Transferencia_Areas
                                ,Pv_Error              => Pv_Error);

   if Pv_Error is not null then
     raise Le_Error;
   end if;


   --Obtener el tamaño de la Trama que contiene el detalle del presupuesto
   Ln_TamañoTrama := length(Pv_Trama);

   --Descomponer trama
   for i in 1..Ln_TamañoTrama loop

     Lv_Caracter := substr(Pv_Trama,i,1);
     if Lv_Caracter != '|' then
       Lv_Palabra := Lv_Palabra||Lv_Caracter;
     else
       Ln_Contador := Ln_Contador+1;
       if Ln_Contador = 1 then
         Lv_CodTipoPresupuesto := Lv_Palabra;

         --Verifica si el tipo de presupuesto que se desea actualizar es válido
         open C_TipoPresupuesto(Lv_CodTipoPresupuesto);
         fetch C_TipoPresupuesto into Lc_TipoPresupuesto;
         Lb_Found := C_TipoPresupuesto%notfound;
         close C_TipoPresupuesto;

         if Lb_Found then
           Pv_Error := 'El tipo de presupuesto que desea registrar no es válido';
           raise Le_Error;
         end if;

         Lv_Palabra := null;

       else
         Ln_ValorPresupuesto := to_number(replace(Lv_Palabra,'.',','));

         --En caso de Asignar
         if Pv_Tipo = 'A' then --Asignación
           --Validar que el monto a asignar no sea mayor al valor disponible
           open C_DatosPsi (Lv_CodTipoPresupuesto);
           fetch C_DatosPsi into Lc_DatosPsi;
           Lb_Found := C_DatosPsi%notfound;
```

```
          close C_DatosPsi;

          if Lb_Found then
            Pv_Error := 'No existe registro del presupuesto que desea asignar';
          end if;

          if Lc_DatosPsi.Saldo < Ln_ValorPresupuesto then
            Pv_Error := 'No hay disponible en el presupuesto de Sistemas el valor que
                desea Asignar';
            raise Le_Error;
          end if;

        end if;

        --Si el valor a asignar o a devolver es cero, no debe registrar
        if Ln_ValorPresupuesto > 0 then

          --Registrar la devolución o la asignación
          CYPK_OBJ_PRESUPUESTOS.P_INS_MOVIMIENTO_PRESUPUESTO
(Pn_CodigoProyecto  => to_number(Pv_CodigoProyecto)
                                          ,Pn_CodigoPresup    =>
Lc_TipoPresupuesto.Codigo_Presupuesto
                                          ,Pn_Valor           => Ln_ValorPresupuesto
                                          ,Pd_FechaMovimiento => sysdate
                                          ,Pv_CodigoUsuario   => Pv_CodigoUsuario
                                          ,Pv_TipoMov         => Lv_TipoMovimiento
                                          ,Pv_Estado          => 'E'
                                          ,Pn_CodigoMov       =>
Ln_CodigoMovimiento
                                          ,Pv_Error           => Pv_Error);

          --Calcular el nuevo saldo
          if Pv_Tipo = 'D' then --Devolución
            Ln_Saldo :=  Lc_TipoPresupuesto.Saldo + Ln_ValorPresupuesto;
          elsif Pv_Tipo = 'A' then --Asignación
            Ln_Saldo := Lc_TipoPresupuesto.Saldo - Ln_ValorPresupuesto;
          end if;

          --Devolver el monto
          CYPK_OBJ_PRESUPUESTOS.P_UPD_PRESUPUESTO
(Pn_CodigoPresupuesto  => Lc_TipoPresupuesto.Codigo_Presupuesto
                                    ,Pn_ValorInicial      =>
Lc_TipoPresupuesto.Valor_Inicial
                                    ,Pn_saldo             => Ln_Saldo
                                    ,Pn_ValorFinal        =>
Lc_TipoPresupuesto.Valor_Final
                                    ,Pn_TransferenciaAreas =>
Lc_TipoPresupuesto.Transferencia_Areas
```

```
                                    ,Pv_Error                    => Pv_Error);

          if Pv_Error is not null then
            raise Le_Error;
          end if;

        end if;

        --Obtener la suma total
        Ln_ValorTotal := Ln_ValorTotal + Ln_ValorPresupuesto;

        Ln_Contador := 0;
        Lv_Palabra  := null;

      end if;
    end if;

  end loop;

  --Actualizar en la tabla de costos como finalizado
  if Pv_Tipo = 'D' then --Devolución

    if Ln_ValorTotal > to_number(Pv_ValorTotal) then
      Pv_Error := 'La distribución no puede ser mayor al valor a devolver';
      raise Le_Error;
    elsif Ln_ValorTotal < to_number(Pv_ValorTotal) then
      Pv_Error := 'La distribución no puede ser menor al valor a devolver';
      raise Le_Error;
    end if;

    update cyp_costos_proyecto cp
    set    cp.finalizado = 'S'
    where  cp.codigo_proyecto = to_number(Pv_CodigoProyecto);
  end if;

  Pv_Error := 'Transacción Realizada con Exito';
     commit;
exception
  when Le_Error then
    Pv_Error := Lv_NombrePrograma ||' - '|| Pv_Error;
    rollback;
  when others then
    pv_error := Lv_NombrePrograma||' - '||sqlerrm;
    rollback;

end P_TRANSFERENCIA_PROYECTOS;
```

Nombre*:* P_ASIGNA_PRESUPUESTO_PROYECTO

Origen: CYPK_TRX_PRESUPUESTOS

Parámetros: Pv_CodigoProyecto, Pv_CodigoUsuario, Pv_ValorTotal, Pv_Trama Pv_Tipo, Pv_Error

Definición: Se registra aumento de presupuesto en un determinado proyecto.

```
procedure P_ASIGNA_PRESUPUESTO_PROYECTO  (Pn_CodigoProyecto
                                                    in         number
                                  ,Pv_CodigoUsuario        in     varchar2
                                  ,Pv_Error                    out varchar2) is

  ----------------------------------------------------------------------------------------
  -- Purpose   :
  -- Author    : Grupo # 9
  -- Created   : 10/08/2007
  -- Parameter : Pv_Error          Retorna: el mensaje de error, si ocurriese alguno
  ----------------------------------------------------------------------------------------

    cursor C_Proyecto is
    select *
    from   gen_proyectos p
    where  p.codigo_proyecto = Pn_CodigoProyecto
    and    p.codigo_estado = 2; --OJO preguntar cual estado es el que indica que esta
listo para darle el dinero al proyecto

    cursor C_Movimientos is
    select mp.codigo_presupuesto,
           mp.valor,
           mp.tipo_movimiento
    from   cyp_movimiento_presupuesto mp
    where  mp.codigo_proyecto = Pn_CodigoProyecto
    and    mp.estado = 'E';

    cursor C_Presupuestos (Cn_CodigoPresupuestos   number)is
    select p.valor_inicial,
           p.saldo,
           p.valor_final,
           p.transferencia_areas
    from   cyp_presupuesto p
    where  p.codigo_presupuesto = Cn_CodigoPresupuestos
    and    p.estado = 'A';
```

```
    Lc_Proyecto                 C_Proyecto%rowtype;
    Lc_Presupuestos             C_Presupuestos%rowtype;

    Lv_NombrePrograma           varchar2(50);
    Lv_Bandera                  varchar2(1);

    Ln_Saldo                    number;

    Lb_Found                    boolean;

    Le_Error                    exception;

  begin

    Lv_NombrePrograma :=
'CYPK_TRX_PRESUPUESTOS.P_ASIGNA_PRESUPUESTO_PROYECTO';

    open C_Proyecto;
    fetch C_Proyecto into Lc_Proyecto;
    Lb_Found := C_Proyecto%notfound;
    close C_Proyecto;

    if Lb_Found then
      Pv_Error := 'El proyecto no existe o esta en un estado en el que no se le puede
asignar el presupuesto';
      raise Le_Error;
    end if;

    --Actualizar en la tabla de movimientos, pero con estado E
    --lo que indica que se descuenta el dinero del presuspuesto
    CYPK_OBJ_PRESUPUESTOS.P_UPD_MOVIMIENTO_PRESUPUESTO
    (Pn_CodigoProyecto   => Pn_CodigoProyecto
                                      ,Pv_CodigoUsuario    => Pv_CodigoUsuario
                                      ,Pv_Estado           => 'E'
                                      ,Pv_Error            => Pv_Error);


    --Consultar los valores a disminuir
    Lv_Bandera := '0';
    for i in C_Movimientos loop
      Lv_Bandera := '1';

      open C_Presupuestos (i.codigo_presupuesto);
      fetch C_Presupuestos into Lc_Presupuestos;
      Lb_Found := C_Presupuestos%notfound;
      close C_Presupuestos;

      if Lb_Found then
```

```
        Pv_Error := 'Error al intentar obtener el valor inicial del presupesto';
        raise Le_Error;
      end if;

      Ln_Saldo := Lc_Presupuestos.saldo - i.valor;

      --Disminuir del presupuesto de sistemas
      CYPK_OBJ_PRESUPUESTOS.P_UPD_PRESUPUESTO
(Pn_CodigoPresupuesto => i.codigo_presupuesto
                                    ,Pn_ValorInicial      => Lc_Presupuestos.valor_inicial
                                    ,Pn_saldo             => Ln_Saldo
                                    ,Pn_ValorFinal        => Lc_Presupuestos.valor_final
                                    ,Pn_TransferenciaAreas =>
Lc_Presupuestos.Transferencia_Areas
                                    ,Pv_Error             => Pv_Error);

    end loop;

    if Lv_Bandera = '0' then
      Pv_Error := 'No se encontraron los movimientos de presupuestos relacionados
con este proyecto';
      raise Le_Error;
    end if;


  exception
    when Le_Error then
      Pv_Error := Lv_NombrePrograma ||' - '|| Pv_Error;
      rollback;
    when others then
      pv_error := Lv_NombrePrograma||' - '||sqlerrm;
      rollback;

  end P_ASIGNA_PRESUPUESTO_PROYECTO;
```

Nombre*:* P_REGISTRA_COSTOS_RECURSOS

Origen: CYPK_TRX_PRESUPUESTOS

Parámetros: (Pv_CodigoSubTipoServicio, Pv_Valor, Pv_Tipo, Pv_Error

Definición: Permite registrar nuevos valores en los recursos que interviene en un Proyecto.

```
procedure P_REGISTRA_COSTOS_RECURSOS    (Pv_CodigoSubTipoServicio in
                                         varchar2
                              ,Pv_valor             in      varchar2
                              ,Pv_Error               out  varchar2) is

  ---------------------------------------------------------------------------------------
  -- Purpose   :
  -- Author    : Grupo # 9
  -- Created   : 10/08/2007
  -- Parameter : Pv_Error          Retorna: el mensaje de error, si ocurriese alguno
  ---------------------------------------------------------------------------------------

    cursor C_Existe is
    select *
    from   cyp_costos_recursos cr
    where  cr.codigo_sub_tipo_servicio = Pv_CodigoSubTipoServicio
    and    cr.estado = 'A';

    Lv_NombrePrograma        varchar2(90);

    Ln_CodCostoRecurso       number;

    Ln_Valor                 number;

    Lc_Existe                C_Existe%rowtype;

    Lb_Found                 boolean;

    Le_Error                 exception;

  begin

    Lv_NombrePrograma :=
'CYPK_TRX_PRESUPUESTOS.P_REGISTRA_COSTOS_RECURSOS';

    --Verificar si ya existe registros del recurso que se desea grabar
    open C_Existe;
    fetch C_Existe into Lc_Existe;
    Lb_Found := C_Existe%found;
    close C_Existe;
```

```
    if Lb_Found then
      --Poner en estado Finalizado el registro actual
      CYPK_OBJ_PRESUPUESTOS.P_UPD_COSTOS_RECURSOS
(Pn_CodCostoRecurso => Lc_Existe.Codigo_Costo_Recurso
                                      ,Pv_Error          => Pv_Error);

      if Pv_Error is not null then
        raise Le_Error;
      end if;

    end if;

    Ln_Valor := to_number(replace(Pv_valor,'.',','));

    --Insertar el registro del costo
    CYPK_OBJ_PRESUPUESTOS.P_INS_COSTOS_RECURSOS
(Pv_CodigoTipoPart  => Lc_Existe.Codigo_Tipo_Participante
                                    ,Pn_Valor                 => Ln_Valor
                                    ,Pv_Estado                => 'A'
                                    ,Pd_FechaInicio           => sysdate
                                    ,Pd_FechaFin              => null
                                    ,Pv_CodigoSubTipo      => Pv_CodigoSubTipoServicio
                                    ,Pn_CodCostoRecurso => Ln_CodCostoRecurso
                                    ,Pv_Error                 => Pv_Error);

    if Pv_Error is not null then
      raise Le_Error;
    end if;

    commit;

  exception
    when Le_Error then
      Pv_Error := Lv_NombrePrograma ||' - '|| Pv_Error;
      rollback;
    when others then
      pv_error := Lv_NombrePrograma||' - '||sqlerrm;
      rollback;

  end P_REGISTRA_COSTOS_RECURSOS;
```

2. Manual de Usuario

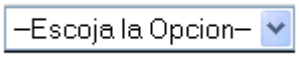
2.1. Ingreso al Modulo de Control de Presupuesto y Costos del Proyecto.

Al ingresar al Modulo, aparecerá una interfaz amigable al usuario, la le permitirá ingresar su usuario y clave posteriormente se dará inicio a la llamada de los procesos propios del modulo. Este proceso de validar el usuario y contraseña es parte de la integración con el Modulo de Seguridades.

2.2. Definiciones Utilizadas en la Aplicación

Se utilizaran definiciones tales como:

- ✓ Botones
- ✓ Mensajes
- ✓ Cajas de información
- ✓ Seleccionar

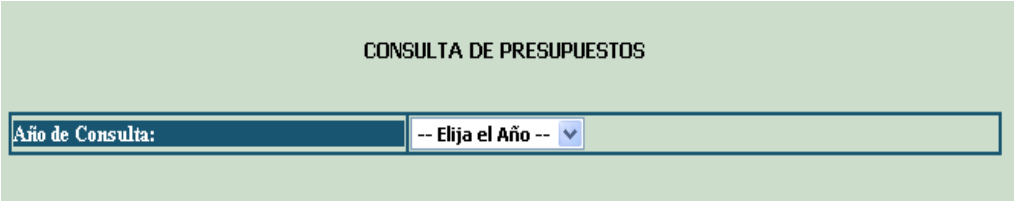
Definición	Descripción
	Este icono significa guardar o Grabar los datos.
	Este icono significa salir de la pantalla actual.
	Este icono significa buscar según el criterio de datos.
	Solo se podrá seleccionar una opción y se mostrara su contenido.
	Abrirá una pantalla y se podrá modificar datos de un registro seleccionados.
	Al hacer clic en la flecha marcada

	con celeste se podrá escoger una opción de la lista.
	Cuando se ingresa y se graba el presupuesto correctamente.
	Cuando se ingresa y se graba el costo de un recurso correctamente.

2.3. Descripción de pantallas.

2.3.1. Administración de Presupuesto (Consulta).

Al hacer clic en este link se mostrara un encabezado que contiene el año del presupuesto a consultar. Después de seleccionar el año deseado, aparece el detalle de distribución del valor del presupuesto en los diferentes tipos de presupuestos que existen.

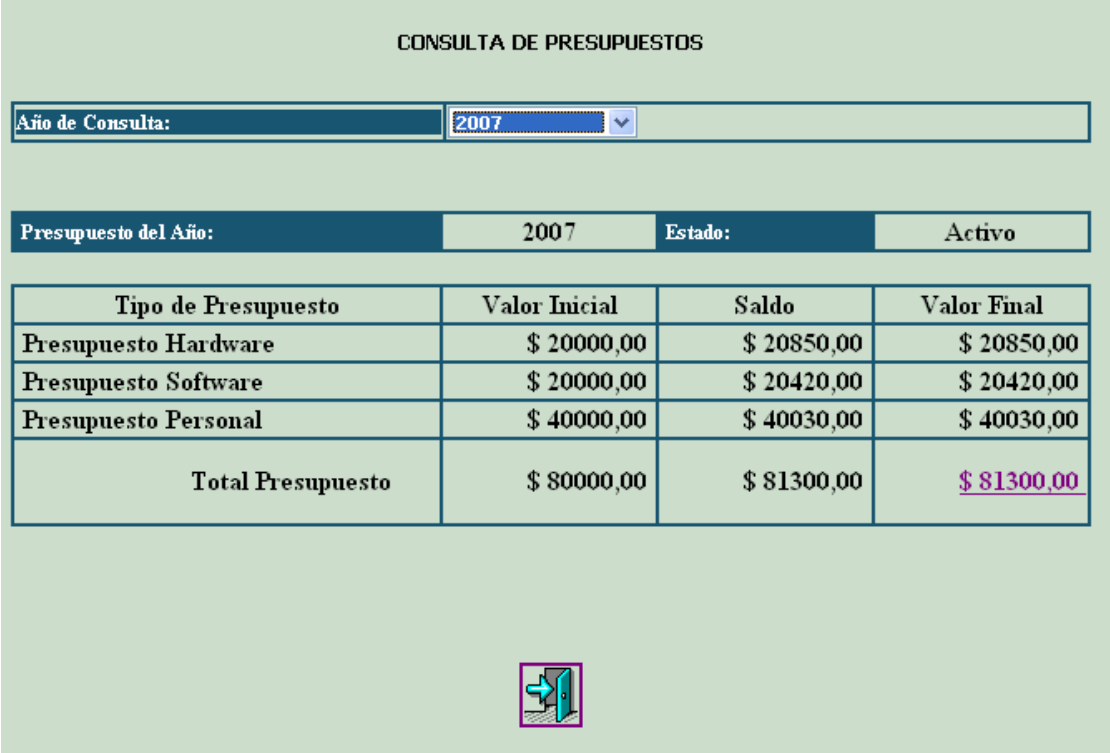


La información que se visualizara será:

Valor inicial: Este valor es asignado al inicio del año a cada tipo presupuesto.

Saldo: El saldo que va quedando por los movimientos que se realizan al asignar parte del presupuesto, debido a los costos que intervienen de un proyecto determinado.

Valor final: Este valor reflejara si cada tipo presupuesto aumento su valor, sea este por transferencia o sobrante de dinero de presupuesto anterior.



CONSULTA DE PRESUPUESTOS

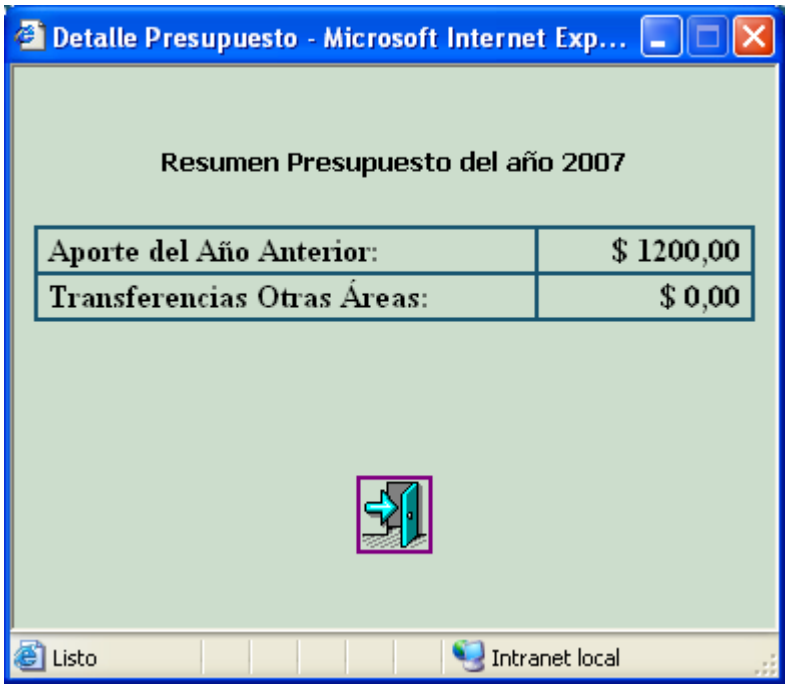
Año de Consulta:	2007

Presupuesto del Año:	2007	Estado:	Activo

Tipo de Presupuesto	Valor Inicial	Saldo	Valor Final
Presupuesto Hardware	$ 20000,00	$ 20850,00	$ 20850,00
Presupuesto Software	$ 20000,00	$ 20420,00	$ 20420,00
Presupuesto Personal	$ 40000,00	$ 40030,00	$ 40030,00
Total Presupuesto	$ 80000,00	$ 81300,00	$ 81300,00

Total Presupuesto: Suma de cada columna presentada. Por lo consiguiente si se desea tener el detalle de transferencias existentes y valores agregados. Se tiene el *total* de valor final

representado por un link que llevara a la pantalla de resumen de estos valores adicionales al presupuesto.



Si desea salir de la pantalla de resumen debe hacer clic en el icono salir.

2.3.2. Administración de Presupuesto (Ingreso).

Esta pantalla permite ingresar el presupuesto del siguiente año. Para grabar los datos ingresados debe hacer clic en el icono del disquete.

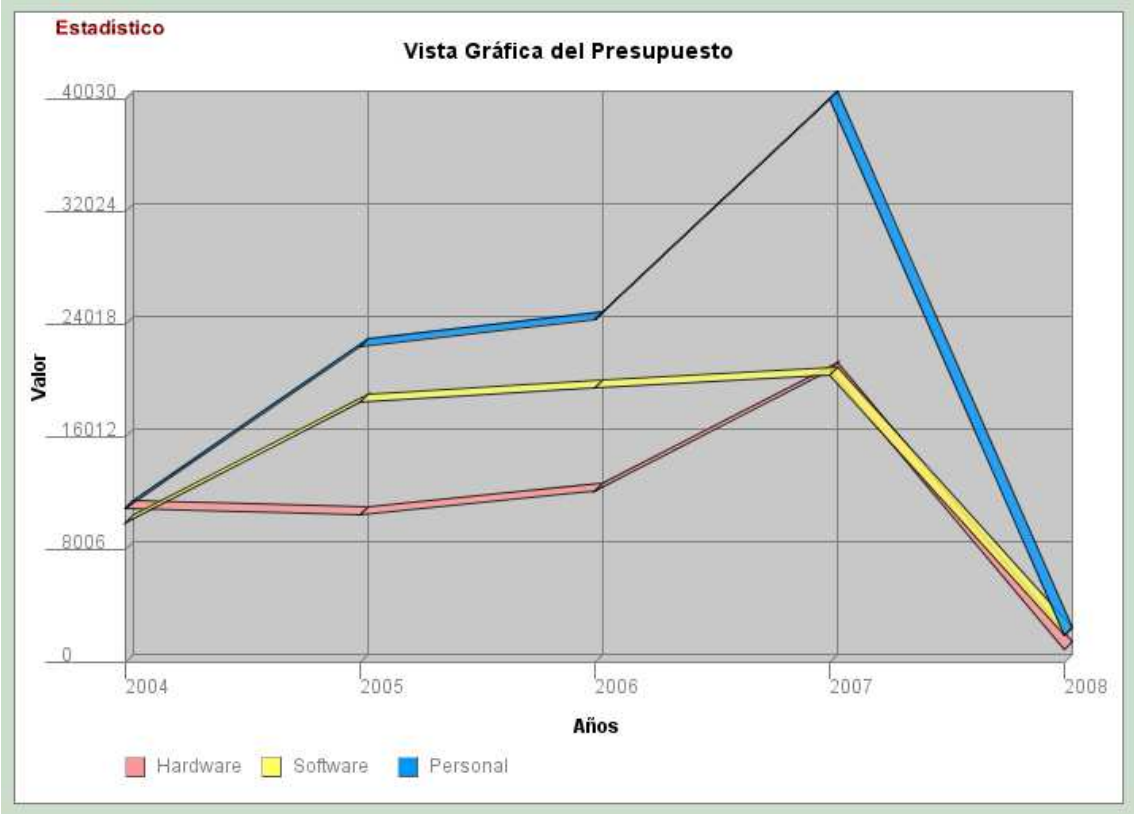
Si desea salir en el otro icono. Los datos se guardan con estado *Próximo.*

2.3.3. Administración de Presupuesto (Vista Grafica)

Esta pantalla visualizara información histórica y actual gráficamente, esto servirá de herramienta para la toma de decisiones al momento de

decidir en realizar su presupuesto para el próximo año, realizar ajustes en cada tipo de presupuestos que este administrando.

Esta información muestra los valores de tipos de presupuestos de los últimos cinco años.



2.3.4. Costos de Proyectos (Costo estimado).

En esta pantalla se encuentran todos lo proyectos que tengan estado cr, ap.

El estado cr, indica que el proyecto tiene realizado ya el cronograma, es decir, puede calcularse el costo estimado.

El estado ap, indica que el proyecto tiene calculado el costo estimado pero esta en espera de aprobación. En esta bandeja usted podrá aprobar el costo estimado calculado a un determinado proyecto.

Usted podrá utilizar criterios de búsqueda por área, estado y adicionalmente ingresar rangos de fechas. Para ejecutar la búsqueda presione el icono a continuación se mostrara su información.

BANDEJA DE COSTOS ESTIMADOS

Cod. Proyecto	Nombre Proyecto	Area Solicita	Fecha Solicita	Prioridad	Líder
1	[11] -Reporte De Facturas	Tecnologia	24/08/2007	Rqemalta	No Asignado
2	[21] - Generacion De Codigos En Factura	Tecnologia	24/08/2007	Rqemalta	No Asignado
3	[101] - Aplicacion De Retencion De Iva	Financiera	31/08/2007	Novalue	No Asignado
4	[121] - Reporte De Ventas De Abril	Financiera	31/08/2007	Rqemalta	No Asignado

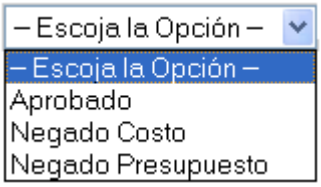
Para aprobar el costo estimado de un proyecto se debe hacer clic en el código del proyecto que aparece como links, esto llevara a una pantalla que contiene toda la información necesaria para poder aprobar el costo estimado.

Para decidir si se aprueba o no el costo estimado de un proyecto se utiliza un combo que contiene tres opciones y cada opción deberá ser respaldada con la debida justificación. A continuación las siguientes opciones:

Aprobado.- El costo estimado se lo aprueba, es decir no tiene ningún inconveniente con los valores del cálculo.

Negado Costo.- El valor del cálculo del costo estimado esta muy elevado, es decir no representara el análisis costo beneficio de este.

Negado presupuesto.- No hay presupuesto de sistemas para cubrir con los gastos de este proyecto. Puede tener una salvaguarda con los procesos de transferencias que se manejan en este modulo.



Luego de haber aprobado el costo estimado de un proyecto este podrá asignársele el estado EC, el cual indica que puede empezar su curso normal de tareas. Recibimos la confirmación de esto visualizando un mensaje que indicara que la transacción se realizo sin inconvenientes.

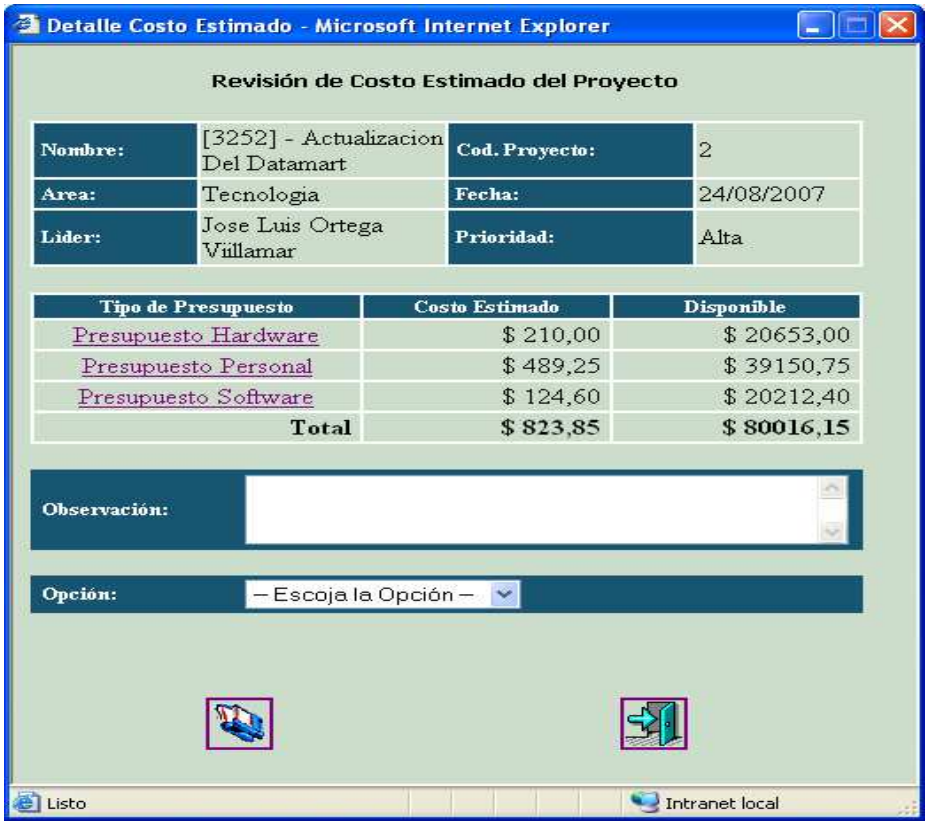
Esta es la pantalla que mostrara la información necesaria para tomar la decisión de aprobar o no el costo estimado de un pre-proyecto determinado.

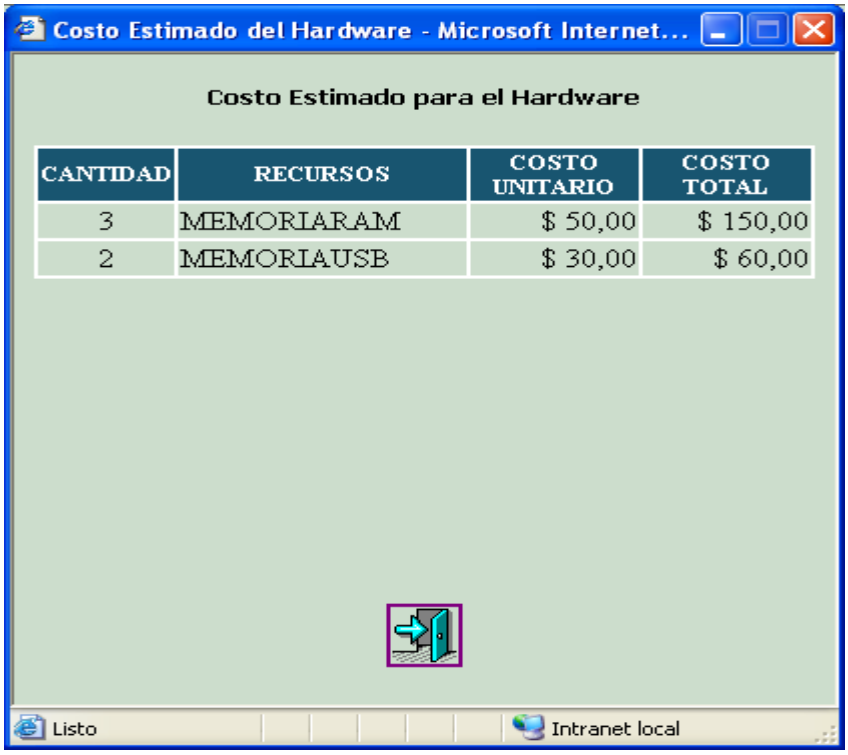
Existen otros links, es decir para cada tipo presupuesto que mostrará la informacion al detalle de cómo es la distribución por cada rubro que interviene. De esta manera Tendremos por ejemplo:

El presupuesto para Hardware será:

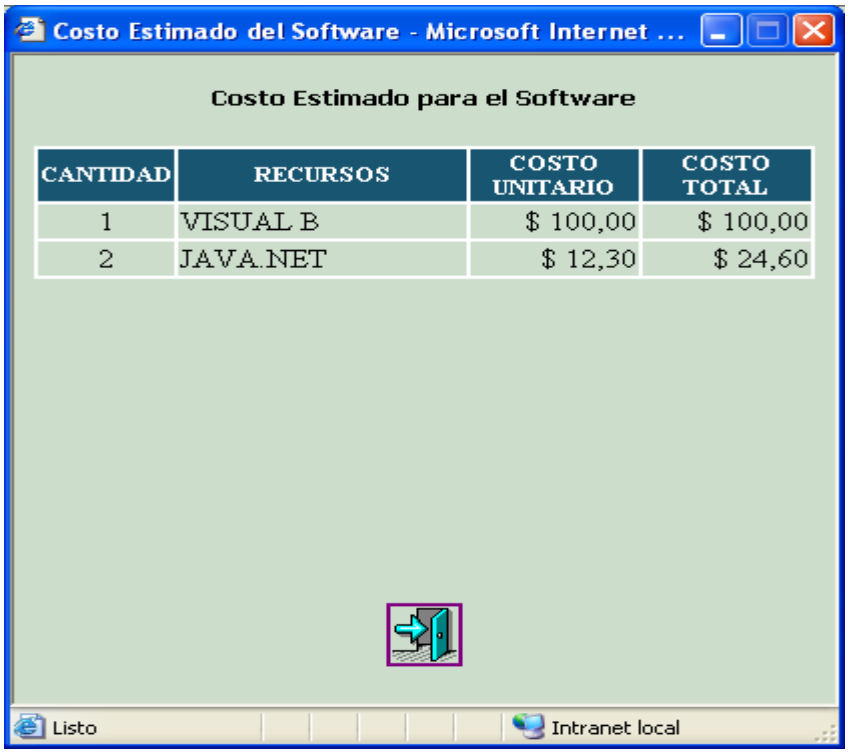
Tres memorias ram a $50,oo y su respectivo costo total.

Por lo tanto cada tipo presupuesto mostrara información de este tipo de acuerdo a la descripción que contengan.


Detalle Costo Estimado - Microsoft Internet Explorer
Revisión de Costo Estimado del Proyecto
Nombre:
[3252] - Actualizacion Del Datamart
Cod. Proyecto:
2
Area:
Tecnologia
Fecha:
24/08/2007
Lider:
Jose Luis Ortega Viillamar
Prioridad:
Alta
Tipo de Presupuesto
Costo Estimado
Disponible
Presupuesto Hardware
$ 210,00
$ 20653,00
Presupuesto Personal
$ 489,25
$ 39150,75
Presupuesto Software
$ 124,60
$ 20212,40
Total
$ 823,85
$ 80016,15
Observación:
Opción:
– Escoja la Opción –
Listo
Intranet local


Costo Estimado del Hardware - Microsoft Internet...
Costo Estimado para el Hardware
CANTIDAD
RECURSOS
COSTO UNITARIO
COSTO TOTAL
3
MEMORIARAM
$ 50,00
$ 150,00
2
MEMORIAUSB
$ 30,00
$ 60,00
Listo
Intranet local

Costo Estimado del Recurso Humano - Microsoft I...
Costo Estimado para el Personal
RECURSOS
COSTO/DIAS
TIEMPO/DIAS
COSTO TOTAL
Recurso 1
$ 25,75
11
$ 283,25
Recurso 2
$ 25,75
8
$ 206,00
Listo
Intranet local



2.3.5. Costos de Proyectos (Revisión Costo real - Bandeja)

Esta pantalla permite realizar búsquedas. Usted debe seleccionar el área de la lista desplegable, de la misma forma seleccionar el estado del proyecto y luego ingresar el rango de fecha que desee la información a mostrar. Para empezar la búsqueda debe hacer clic en el icono en forma de lupa.

La información que se mostrara a continuación será el código del proyecto, Nombre del proyecto, área a la que pertenece el proyecto, líder, estado, fecha inicio, fecha fin, prioridad.

Estado:

Proyecto en curso: indica que el proyecto tuvo aprobación de presupuesto de acuerdo con el cálculo del costo estimado.

Proyecto en stand by, indica que esta sujeto a la gestión de dinero sea este por aprobación de transferencia entre áreas o por otro caso, debido a que se quedo sin presupuesto para seguir con su flujo normal.

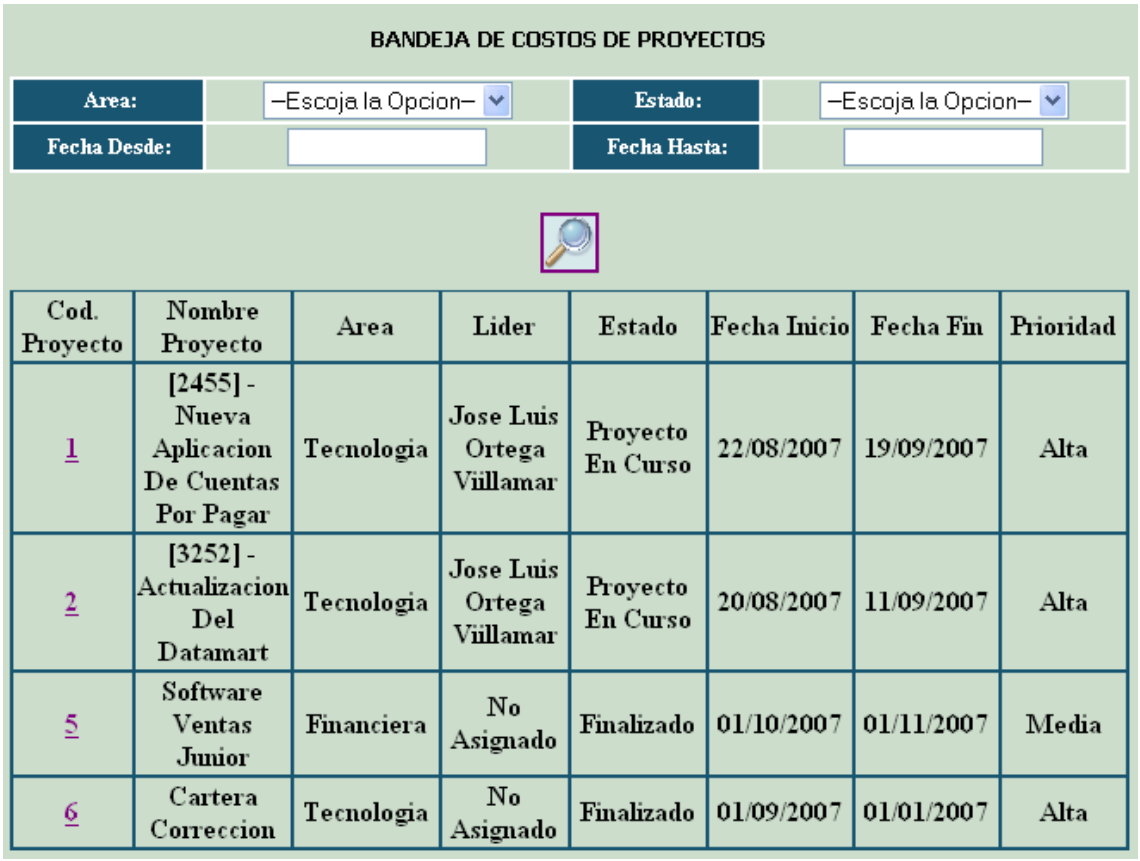
Proyecto finalizado, indica que esta sujeto a la aprobación de transferencia entre áreas, debido a que se quedo sin presupuesto para seguir con su flujo normal.

Prioridad

Alta: el proyecto es un requerimiento muy importante para el área que solicita.

Media: proyecto se considera un requerimiento de mediana importancia para el área que solicita, o el administrador que tome la decisión.

Baja: El proyecto es un requerimiento con un porcentaje bajo de prioridad, es decir, todavía puede esperar.



BANDEJA DE COSTOS DE PROYECTOS

Area:	–Escoja la Opcion–	Estado:	–Escoja la Opcion–
Fecha Desde:		Fecha Hasta:	

Cod. Proyecto	Nombre Proyecto	Area	Lider	Estado	Fecha Inicio	Fecha Fin	Prioridad
1	[2455] - Nueva Aplicacion De Cuentas Por Pagar	Tecnologia	Jose Luis Ortega Viillamar	Proyecto En Curso	22/08/2007	19/09/2007	Alta
2	[3252] - Actualizacion Del Datamart	Tecnologia	Jose Luis Ortega Viillamar	Proyecto En Curso	20/08/2007	11/09/2007	Alta
5	Software Ventas Junior	Financiera	No Asignado	Finalizado	01/10/2007	01/11/2007	Media
6	Cartera Correccion	Tecnologia	No Asignado	Finalizado	01/09/2007	01/01/2007	Alta

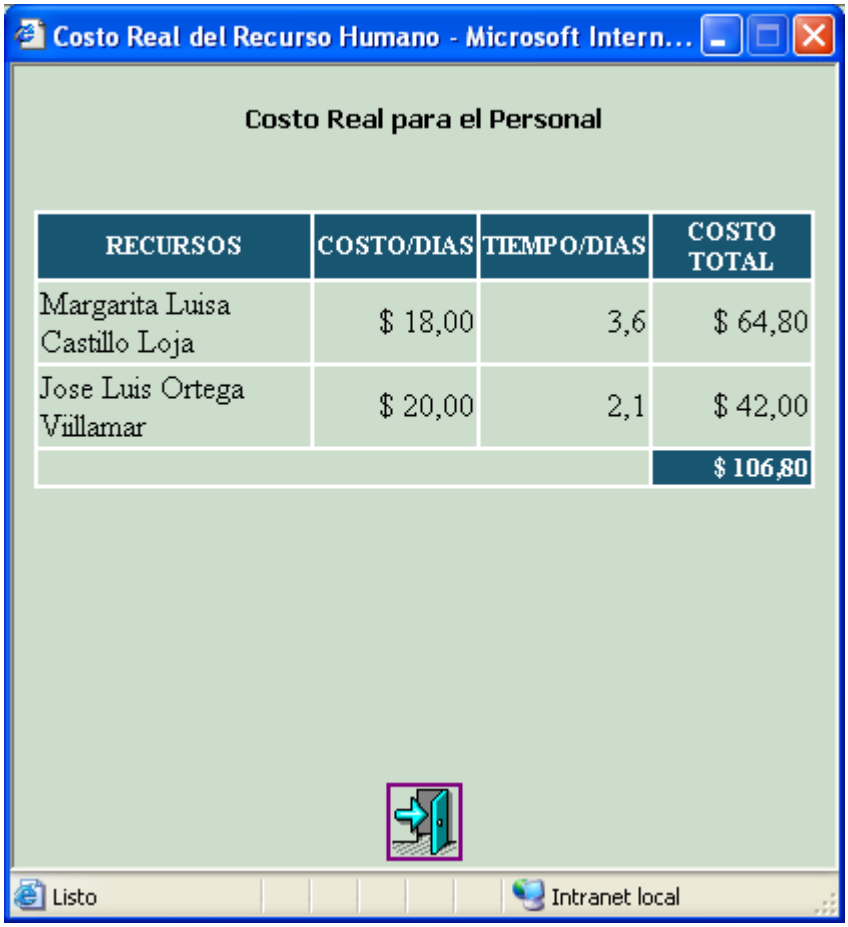
2.3.6. Costos de Proyectos (Revisión Costo real - Detalle).

Si desea conocer más información sobre costos tendrá esta pantalla, creada por cada código del proyecto representado por links, haga clic sobre este y aparece el detalle de cuanto se gasto y cuanto se le asigno a cada tipo de

presupuesto. En la parte superior se visualizara información de cabecera tales como: el nombre del Proyecto, el nombre del área, el líder encargado de este proyecto, el código del proyecto, fecha de inicio del proyecto, y el grado de prioridad asignada, este estado es asignado por el departamento de organización y métodos de la empresa.

Cada tipo de presupuesto incluye información aún mas detallada, como la cantidad de recursos utilizados, descripción del recurso, costo unitario y costo total. Por lo tanto estos campos aparecerán en el Presupuesto de Hardware, Software, recurso Humano,..., etc.

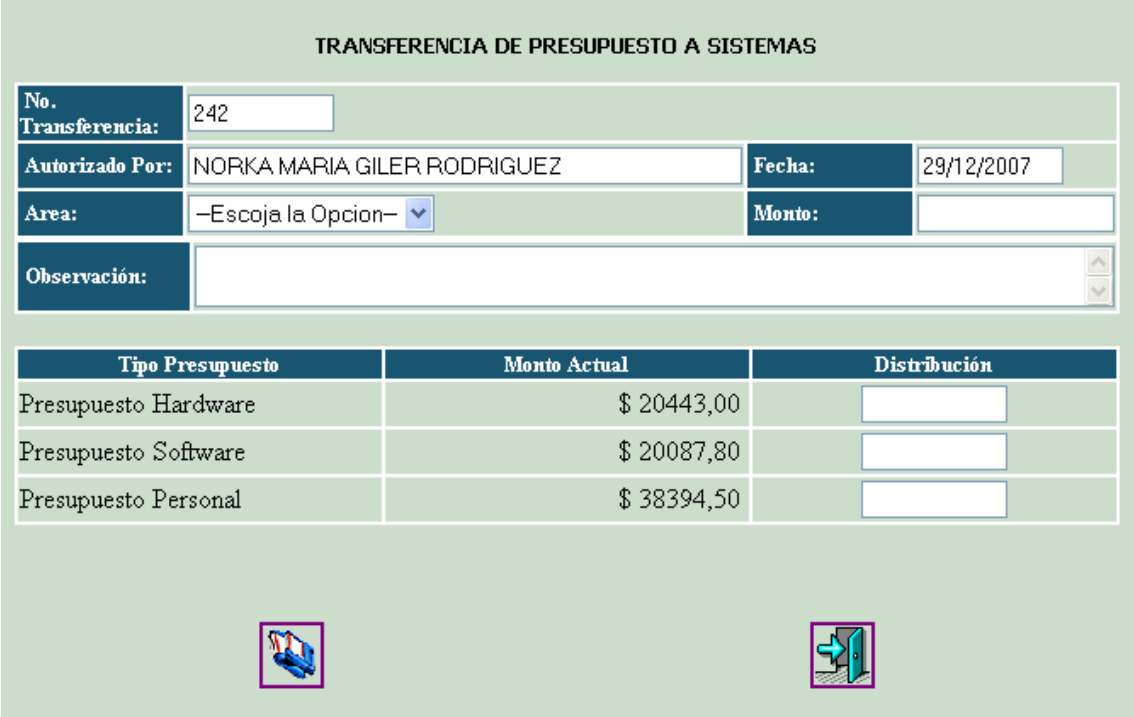
Costo Real del Software - Microsoft Internet Explo...
Costo Real para el Software
CANTIDAD
RECURSOS
COSTO UNITARIO
COSTO TOTAL
1
Licencia Windows
$ 400,00
$ 400,00
$ 400,00
Listo
Intranet local



2.3.7. Transferencias (Área)

Las transferencias nacen de la falta de presupuesto para continuar con un proyecto, entonces para gestionar ese dinero se toma como alternativa el área que ha solicitado el proyecto, ya que se conoce como política de una empresa, que se cuenta con un porcentaje destinado a proyectos.

Este proceso anticipadamente incluye seguridades por lo tanto usted debe ser un usuario con privilegios. Se mostrara en la etiqueta Autorizado por, elija el área origen de la transferencia y escriba la cantidad a ser transferida.



Es muy importante que escriba una observación para esta transferencia, para que sirva de respaldo para el departamento financiero. Además de ingresar la cantidad a transferirse, Usted podrá distribuir esta cantidad entre los diferentes tipos de presupuesto que existan.

Para guardar la información haga clic en el icono grabar.

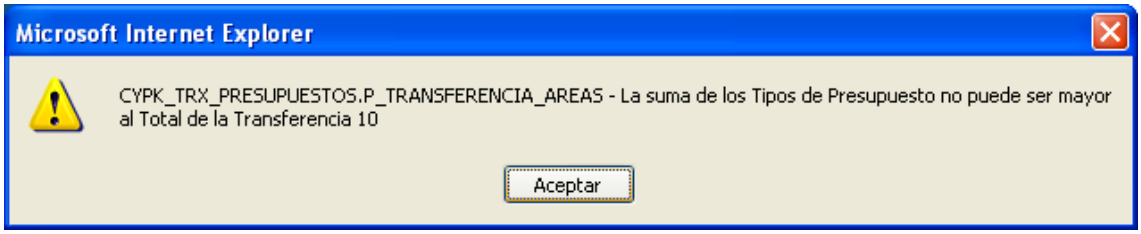
Para salir de la pantalla clic en el icono salir.

Hay tres campos muy importantes para este proceso es decir deben ser llenados o seleccionamos según este lo permita. De lo contrario se le notificara con mensajes enviados por el sistema.

Estos campos son el área que transfiere, si no la selecciona se envía el mensaje de alerta. El monto o cantidad de dinero a transferir, que debe ser ingresada por el usuario. Por ultimo la observación bien detallada del justificativo de la transferencia a realizarse, esta me permitirá tener respaldo de esta decisión a nivel de reportes gerenciales.

Luego de presionar el botón de grabar se le notificara a través de un mensaje que la transacción fue realizada exitosamente.

Si Usted distribuye a cada tipo de presupuesto un valor mayor al monto que esta registrando se le notificara inmediatamente a través de un mensaje de alerta. Esto le permitirá corregir cada valor y ponerlo equivalente al valor del monto.

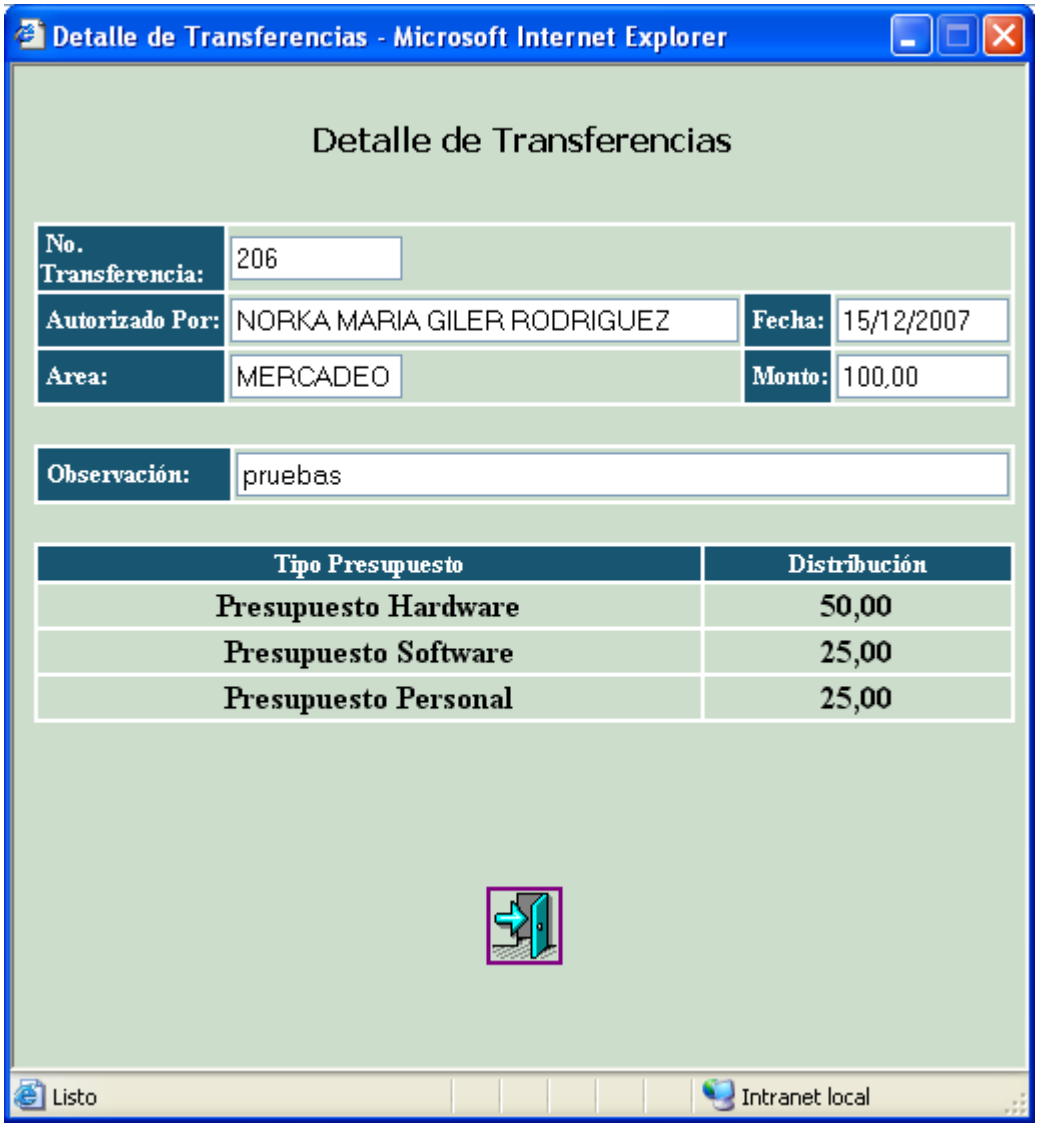

Microsoft Internet Explorer
CYPK_TRX_PRESUPUESTOS.P_TRANSFERENCIA_AREAS - La suma de los Tipos de Presupuesto no puede ser mayor al Total de la Transferencia 10
Aceptar

2.3.8. Transferencias (Consulta).

En esta pantalla permite obtener todas las últimas transferencias realizadas de tal manera, que se podrán visualizar a través de un link que proporcionara una pantalla la fecha en que se realizo, quien la autorizo, cantidad de dinero transferida y cuanto fue asignado a cada tipo presupuesto, área de donde proviene el dinero.

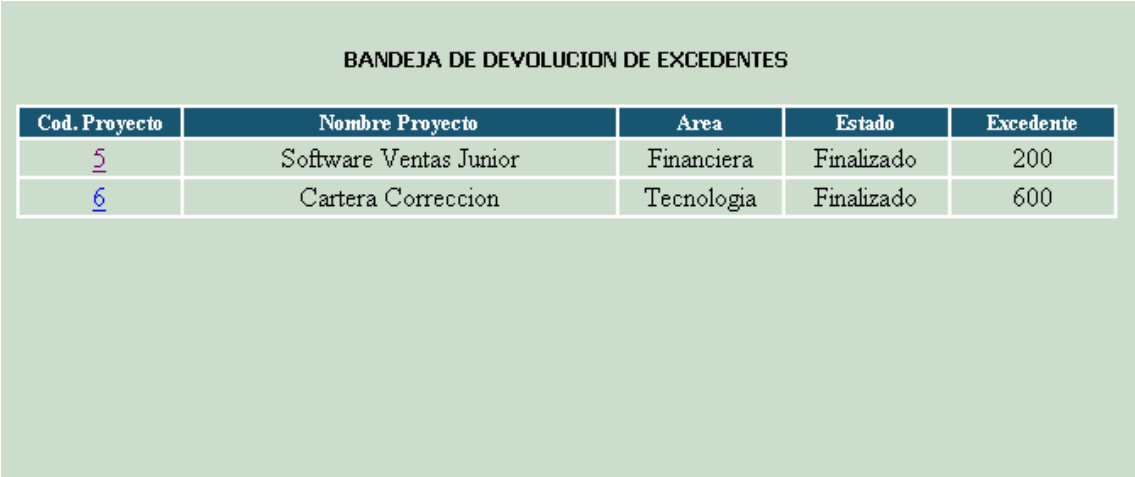
CONSULTA DE TRANSFERENCIAS ENTRE AREAS

Cod. Transferencia	Area Transfiere	Valor Transfiere	Nombre Autoriza	Fecha Transferencia
205	Mercadeo	300,00	Norka Maria Giler Rodriguez	15/12/2007
206	Mercadeo	100,00	Norka Maria Giler Rodriguez	15/12/2007
207	Financiera	100,00	Norka Maria Giler Rodriguez	15/12/2007
231	Financiera	1,00	Norka Maria Giler Rodriguez	20/12/2007
232	Financiera	1,00	Norka Maria Giler Rodriguez	20/12/2007
233	Financiera	1,00	Norka Maria Giler Rodriguez	20/12/2007
237	Financiera	3,00	Norka Maria Giler Rodriguez	20/12/2007
240	Financiera	1,00	Norka Maria Giler Rodriguez	20/12/2007
242	Mercadeo	10,00	Norka Maria Giler Rodriguez	29/12/2007


Detalle de Transferencias - Microsoft Internet Explorer
Detalle de Transferencias
No. Transferencia:
206
Autorizado Por:
NORKA MARIA GILER RODRIGUEZ
Fecha:
15/12/2007
Area:
MERCADEO
Monto:
100,00
Observación:
pruebas
Tipo Presupuesto
Distribución
Presupuesto Hardware
50,00
Presupuesto Software
25,00
Presupuesto Personal
25,00
Listo
Intranet local

2.3.9. Ajustes Costos de Proyectos (Devolución).

Se mostrara una bandeja de todos los proyectos finalizados que tengan sobrante de presupuestos. Para acceder a la pantalla que permite realizar la devolucion se debe hacer clic en el link del codigo del proyecto.



BANDEJA DE DEVOLUCION DE EXCEDENTES

Cod. Proyecto	Nombre Proyecto	Area	Estado	Excedente
5	Software Ventas Junior	Financiera	Finalizado	200
6	Cartera Correccion	Tecnologia	Finalizado	600

Esta pantalla permite registrar dinero sobrante, de todos los proyectos que hayan finalizados. Para realizar esta devolución se debe ingresar el código del proyecto, el monto a devolver, area a la cual pertenece el proyecto,

La devolución de excedentes implica realizar una transferencia de dinero de un proyecto finalizado al presupuesto de sistemas, con esto se logra invertirlos en otros proyectos, es decir, la correcta y completa administración

del presupuesto. Por lo tanto la administración del presupuesto de sistemas será eficiente

2.4.0. Ajustes Costos de Proyectos (Asignación).

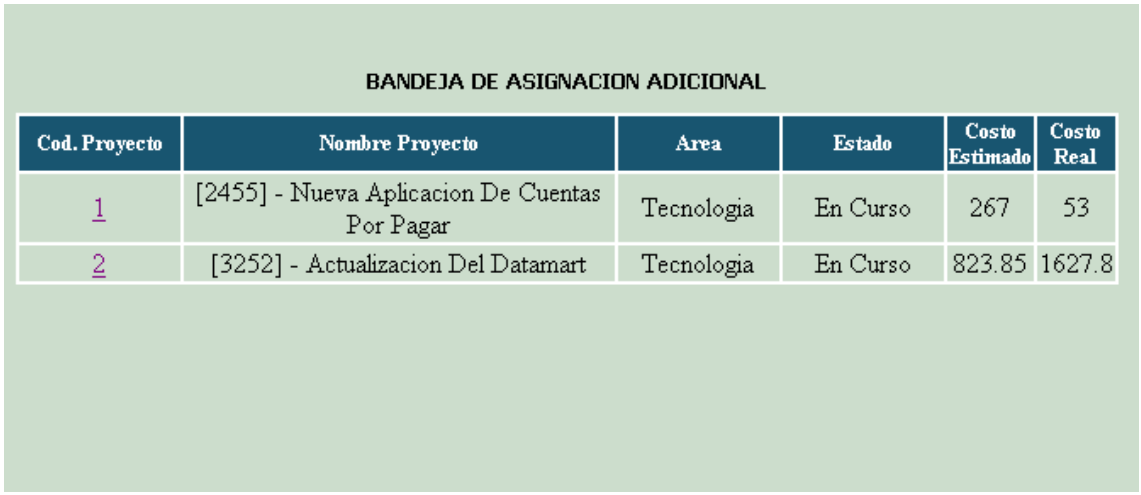
La primera parte de este proceso mostrara una bandeja de proyectos que han solicitado o que necesitan presupuesto para continuar con la ejecución

de sus actividades. De esta pantalla podrás elegir el proyecto que le vas asignar mas dinero del presupuesto de sistemas.

Una vez elegido el proyecto para la asignación se presentara la pantalla la misma que ingresaras el monto a transferir. Para esto se necesitara ingresar monto a asignar entre los campos mas importantes.

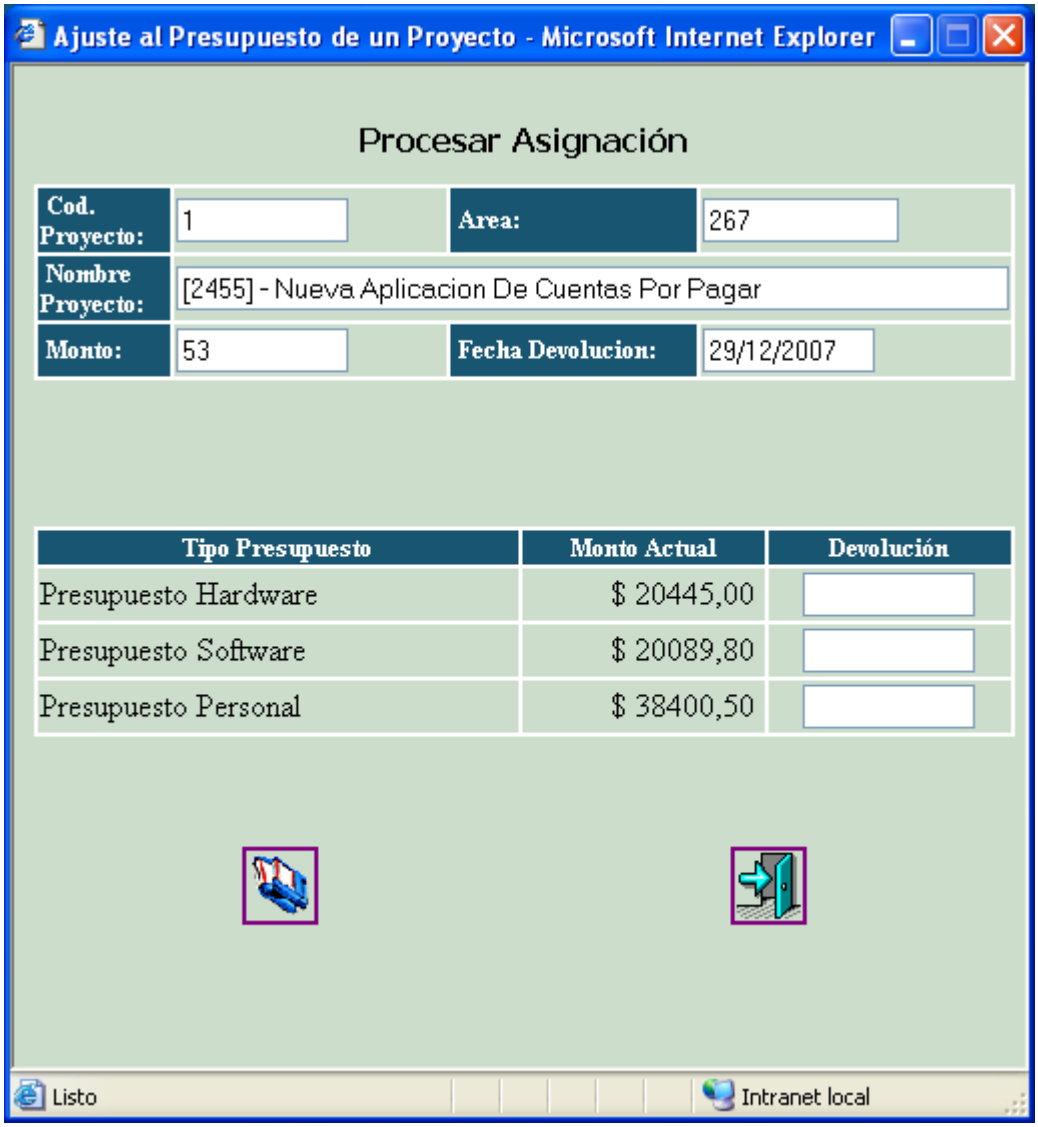
Si desea grabar los cambios haga clic en el icono de grabar.

Si desea salir los cambios haga clic en el icono Salir.



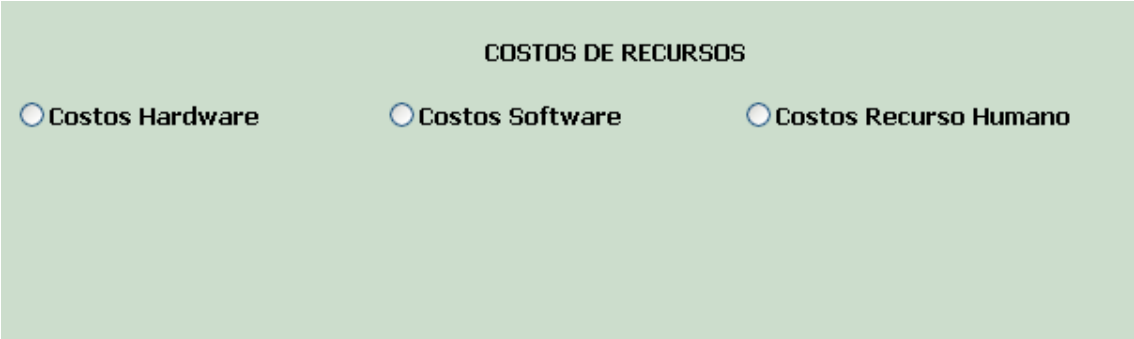
BANDEJA DE ASIGNACION ADICIONAL

Cod. Proyecto	Nombre Proyecto	Area	Estado	Costo Estimado	Costo Real
1	[2455] - Nueva Aplicacion De Cuentas Por Pagar	Tecnologia	En Curso	267	53
2	[3252] - Actualizacion Del Datamart	Tecnologia	En Curso	823.85	1627.8

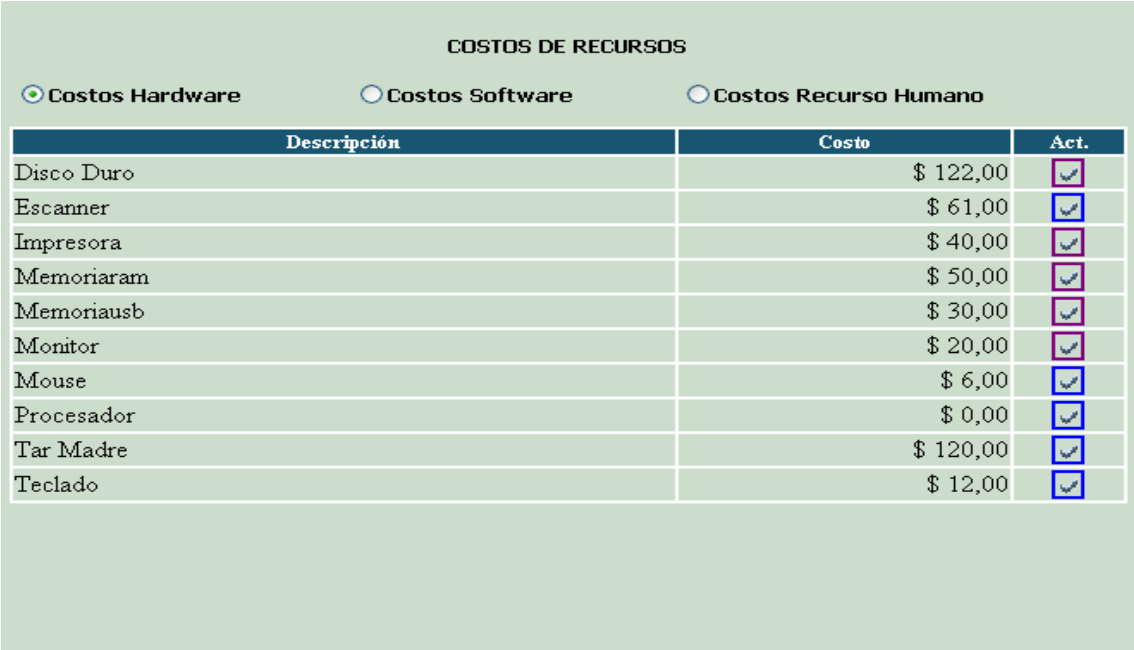


2.4.1. Parametrizaciones (Costos de Recursos).

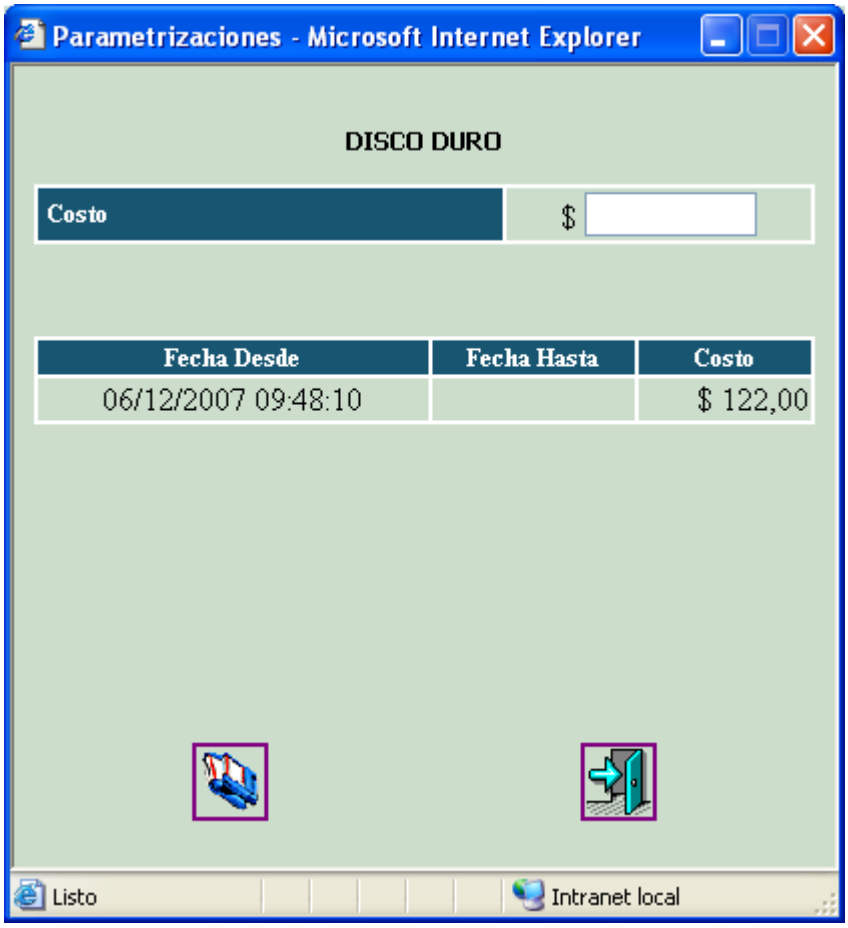
Esta opción permite el ingreso de información de hardware, esta información que se visualiza es alimentada por otros módulos pertenecientes a la gestión de proyectos informáticos.



Usted ha elegido *Costos de Hardware* aparecerá un listado con la descripción del hardware y su respectivo costo. Además incluye una columna con un icono, el mismo que dará acceso a una pantalla en la cual podremos ingresar nuevos costos.



Descripción	Costo	Act.
Disco Duro	$ 122,00	
Escanner	$ 61,00	
Impresora	$ 40,00	
Memoriaram	$ 50,00	
Memoriausb	$ 30,00	
Monitor	$ 20,00	
Mouse	$ 6,00	
Procesador	$ 0,00	
Tar Madre	$ 120,00	
Teclado	$ 12,00	

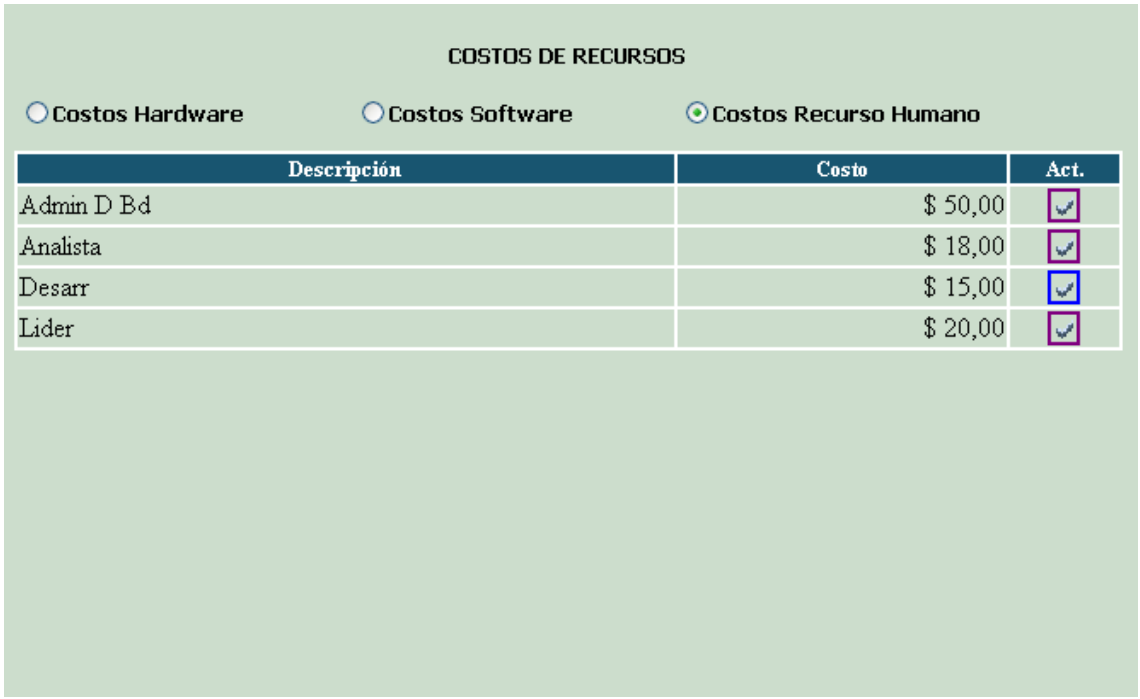


Al ingresar los nuevos costos y presionar grabar se registrara la fecha automáticamente y en el listado aparecerá el nuevo valor ingresado. Por otra parte su transacción queda registrada satisfactoriamente y usted podrá tener la confirmación de un mensaje enviado por el sistema.

Si desea guardar los cambios clic en el icono grabar.

Si desea salir de la pantalla, es decir cancela lo que esta realizando haga clic en el icono salir.

Las siguientes opciones tales como *Costos de Software, Costos de recursos* para actualizarles el valor de cada producto deben seguir los pasos anteriores.


COSTOS DE RECURSOS
Costos Hardware
Costos Software
Costos Recurso Humano
Descripción
Costo
Act.
Admin D Bd
$ 50,00
Analista
$ 18,00
Desarr
$ 15,00
Lider
$ 20,00